2016-09

Optimizing search patterns for multiple searchers prosecuting a single contact in the South China Sea

Lukens, Zachary C.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

OPTIMIZING SEARCH PATTERNS FOR MULTIPLE SEARCHERS PROSECUTING A SINGLE CONTACT IN THE SOUTH CHINA SEA

by

Zachary C. Lukens

September 2016

Thesis Advisor: Hong Zhou
Second Reader: Michael Atkinson

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2016	Master's Thesis 09-11-2014 to 09-23-2016

4. TITLE AND SUBTITLE

OPTIMIZING SEARCH PATTERNS FOR MULTIPLE SEARCHERS PROSECUTING A SINGLE CONTACT IN THE SOUTH CHINA SEA

5. FUNDING NUMBERS

6. AUTHOR(S)

Zachary C. Lukens

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Search and detection theory encapsulates a broad range of scenarios, from searching for lost car keys in a parking lot to prosecuting a submarine in the South China Sea. This research draws on oceanographic properties to develop a search radii for two surface ships, searching for a submarine at various speeds, utilizing one of three search patterns: in-line spiral search, in-line ladder search, and a multi-path ladder search. Analysis reveals which parameters yield the highest probability of detection in a 14nm by 14nm search area based on the oceanographic properties at 21N 119E.

14. SUBJECT TERMS

Search Theory, Undersea Warfare, South China Sea, Anti-Submarine Warfare

15. NUMBER OF PAGES 139

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

OPTIMIZING SEARCH PATTERNS FOR MULTIPLE SEARCHERS PROSECUTING A SINGLE CONTACT IN THE SOUTH CHINA SEA

Zachary C. Lukens
Lieutenant, United States Navy
B.S., Jacksonville University, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2016

Approved by: Hong Zhou
Thesis Advisor

Michael Atkinson
Second Reader

Craig Rasmussen
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Search and detection theory encapsulates a broad range of scenarios, from searching for lost car keys in a parking lot to prosecuting a submarine in the South China Sea. This research draws on oceanographic properties to develop a search radii for two surface ships, searching for a submarine at various speeds, utilizing one of three search patterns: in-line spiral search, in-line ladder search, and a multi-path ladder search. Analysis reveals which parameters yield the highest probability of detection in a 14nm by 14nm search area based on the oceanographic properties at 21N 119E.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to take a moment to thank the U.S. Navy for allowing me the opportunity to attend NPS and pursue my master's degree.

I thank my wife, Heather, for her help during this whole process and for supporting my long hours spent studying and working while raising two children.

I thank Dr. Hong Zhou for serving as my thesis advisor, Dr. Mike Atkinson for being my second reader, and Aileen Houston for being my thesis processor.

I would also like to thank Professor John Joseph, Dr. Daphne Kapolka, and Dr. Ben Reeder for their expertise in undersea warfare, and for providing the critical information from GDEM and BELLHOP to assist in making my thesis a success.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Have you ever dropped your keys in the parking lot? Did you use a systematic approach of walking up and down the aisles to find them, or randomly walk around in the hopes you would stumble upon them? If so, then you are already familiar with the science of Search and Detection Theory (SDT). While the applications for SDT extend to any facet of a "searcher" seeking a "target," this research focuses on a scenario of two surface ships conducting Anti-Submarine Warfare (ASW) in the South China Sea [1].

With recent developments in the South China Sea, the United States Navy has taken a vested interest in Chinese claims of territorial waters extending well beyond those recognized by the international rules of law [2]. China's submarine force increases at a rapid pace, threatening to outnumber our own, and as such the United States grows increasingly dependent on different assets such as surface ships and Autonomous Underwater/Surface Vehicles (AUVs/ASVs) to detect, track, and classify submarines in the area.

1.1 History

SDT originated in World War II (WWII) and represented the birth of operations analysis. Specifically employed for finding a systematic approach to ASW and finding submarines, SDT incorporates all aspects of:

> Stochastic processes, characterization of detection devices, use and interpretation of sweep widths and lateral range curves, true range curves, measures of effectiveness of search-detection systems, allocation of search efforts, sequential search, models of surveillance fields, barriers, tracking, and trailing [1].

Search theory can best be described as

> how to distribute your resources to most efficiently find something when you don't know where it is, but you have some idea about where it might be and how it moves [3].

In the beginning of 1942, the US Navy "gathered elite mathematicians to form the Operational Evaluations Group" [3]. Among these mathematicians was French-born Bernard Koopman, who aided in developing theories of probability among various concepts designed specifically with the idea of locating enemy submarines. Utilizing Bayesian methods, U-boats suffered heavy losses in the Bay of Biscay as a result of the search methods developed [3].

After WWII, search theory was applied to locating lost hydrogen bombs in 1966, missing US submarines in 1968, and extended to locating Soviet ballistic missile submarines throughout the Cold War. In 1974, the US Coast Guard relied on Bayesian models to improve search and rescue operations [3].

The following research focuses on the aspects of stochastic processes, use and interpretation of sweep widths, and tracking modeling, specifically in a naval application.

Historically, detection and location of the enemy facilitates one of the most important functions of any naval operation. Three main aspects comprise the problem of search and detection. The first concerns "the properties of the instrument of detection: the properties of the eye, the characteristics of the radar set, or the nature and capabilities of the sonar equipment, and similarly for any other mechanism of detection which it is proposed to employ" [4]. The second aspect concerns the movement of the searcher, and the motion (if any) of the target relative to the searcher. The final aspect consists of "force requirements and their economy." Without a limit on available forces, the probability of detection increases to certainty. However, since an infinite amount of search assets does not exist, the goal becomes "achieving the required effect with the greatest economy of forces" [4].

1.2 Applications and Previous Research

The application of SDT extends to multiple scenarios, limited only by the imagination of the analyst utilizing it. While it is exercised in the form of a naval problem for this research, many research papers investigate its use for different branches of the military as well.

1.2.1 Applications

SDT is not just a Navy problem. It is applicable to all services and Joint Warfare efforts such as: [1]

- Land surveillance
- Artillery coverage
- Joint ISR
- Drug/Terrorist/Guerrilla Interdiction
- Search and Rescue
- Scouting
- Theater Ballistic Missile Defense
- Joint Fires

If a searcher utilizes a biological or electronic system to attempt to locate a goal (known as a "target"), then the application of an SDT model is appropriate.

1.2.2 Previous Research

Volkan Sozen's 2014 thesis investigated SDT applied to Unmanned Arial Vehicles (UAV's) in a border patrol capactiy [5]. His research focused on a straight-line search area searched in its entirety by one UAV, by both UAV's, and also split into a smaller search area that was then shared by both UAV's. His results showed that probability of detection increased with an extra searcher, and that by splitting the two searchers into two separate areas the border was covered more efficiently.

Dr. Michael Atkinson's research, along with Moshe Kress and Rutger-Jan Lange, investigated searching for a target in one of "n" locations, basing the searcher's ability to execute a search not on movement, but on receipt of intelligence and cost incurred based on searching a certain area [6]. The problem introduced became whether to search an area based on current intelligence at the risk of missing the target, or wait for more intelligence at the risk of the target escaping. The model applied directly to many military, homeland security, and criminal scenarios. The results concluded that if the reliability of the information was high, then waiting to attack was the preferred route, and if an attack was imminent, then searching with less intelligence proved to be more cost efficient.

In [7], H. Zhou and H. Wang investigated the optimal path for a single searcher searching for a single moving target. The paths studied consisted of a loop, a spiral, a square spiral, and a horizontal/vertical "ladder" search. Many of the same principles apply to this research, and the MATLAB codes (introduced later) have been adapted from their Monte Carlo simulations to accommodate two searchers.

The research in [7] demonstrated that if the searcher is stationary and the target is moving, the greatest probability of detection occurs if the searcher is placed in the center. If both searcher and target are moving, then the ladder search provides the highest probability of detection. Expanding upon these results, the following research focuses on two moving searchers and a moving target, and similar results are predicted for the outcome.

CHAPTER 2:
Theory

2.1 General Concepts

The following section outlines the basic principles and mathematics for conducting an exhaustive (ideal) search inside of a pre-defined area.

2.1.1 Terms

Prior to outlining the basis of this research, a brief overview of key terms is required.

Sensor The system being used for detection. In this case, it will be Sound Navigation and Ranging (SONAR).

As The total area of the geographic "box" to be searched. (In square nautical miles.)

vS The velocity of the searcher(s).

vT The velocity of the target.

R The range of the sensor. The sensor will detect any target less than or equal to R, and will not detect anything greater than R. This is commonly referred to as a "cookie-cutter" sensor.

w The sweep-width of the searcher. This is the width of the cookie-cutter footprint. (Note: $w = 2 * R$.)

The cookie cutter sensor is an "ideal" version of a detector, as it perfectly identifies any contact within its range. This idea is not entirely transferable to real-world scenarios, as it does not account for human error, changes in environment that could affect detection, and other unforeseen circumstances that alter the actual critical detection range. However, a cookie cutter sensor provides a simple and practical method to incorporate into a model. [7]

2.1.2 Exhaustive Search

As an introductory scenario, consider a single searcher attempting to cover a fixed area. In terms of sonar, the circular cookie-cutter sensor can detect anything inside of radius R,

which gives it a sweep-width of $2 * R$, as seen in Figure 2.1. Consider that sweep-width moving at velocity v and a rectangular box of "swept" area is created, as seen in Figure 2.2 [1].

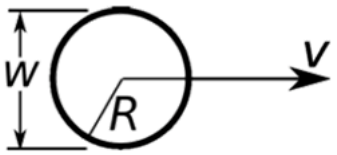


Figure 2.1. Range, Sweepwidth, and Velocity.
Source: [1].

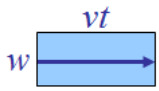


Figure 2.2. Searched Area, Product of Sweepwidth, Velocity, and Time.
Source: [1].

An exhaustive search covers a search area As without overlapping previously searched areas, and without searching outside of borders of the defined area [8]. The most commonly used search methods involve a parallel search (referred to as "ladder" or "lawnmower" searches), and a spiral in/out search, illustrated in Figure 2.3 [1].

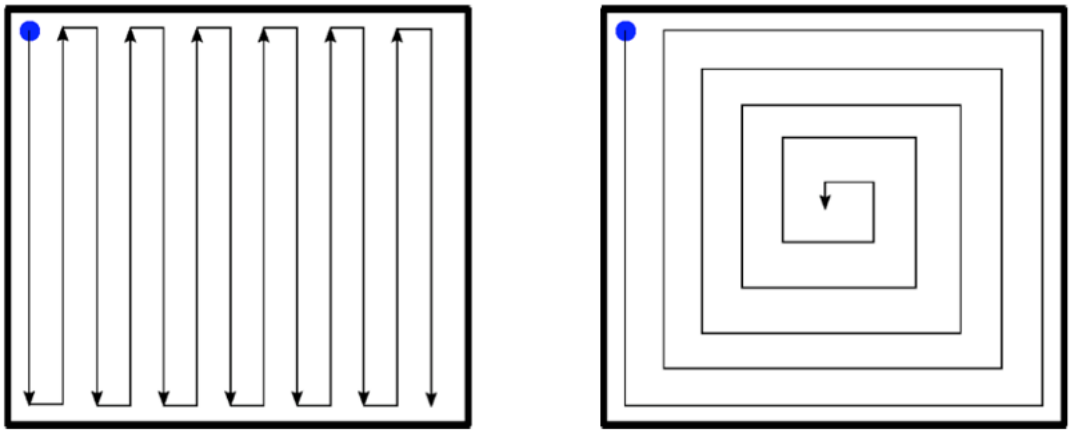


Figure 2.3. A Ladder and Spiral Search.
Source: [1].

As the searcher progresses forward in time "t," the area searched becomes:

$$A(t) = w * (vt) \tag{2.1}$$

For an ideal (or complete) coverage, the time t^* (time to complete coverage) is considered along with velocity and sweepwidth [1].

$$As = v * w * t^* \tag{2.2}$$

After applying simple algebra, the total time to complete coverage then becomes

$$t^* = As/(v * w) \tag{2.3}$$

Again, this t^* value provides the total time to complete coverage in a perfect scenario. In a real-world application, it provides a rough estimate of total time to coverage. (Assuming that overlapping and detection outside of the search area will occur, as well as a sensor that inherently possesses fluctuations in detection range.)

2.1.3 Random Search

Much like the exhaustive search, this simple example of random search incorporates a single searcher with a cookie-cutter sensor seeking a single target. However, random search relaxes the coverage assumptions of exhaustive search by incorporating overlapped search areas, gaps in coverage, and excess coverage areas [1].

Again, consider a searcher with sensor range "R", and sweep width $w = 2 * R$. With a speed of v, in a small amount of time dt the searcher covers an area

$$a = 2 * R * v * dt = w * v * dt.$$

This searched area is equally likely to be anywhere within the overall search area As. So while an actual random search presents itself like that in Figure 2.4, the calculation incorporates throwing n small pieces of searched-area as if it was confetti onto the overall

search area As, as seen in Figure 2.5.

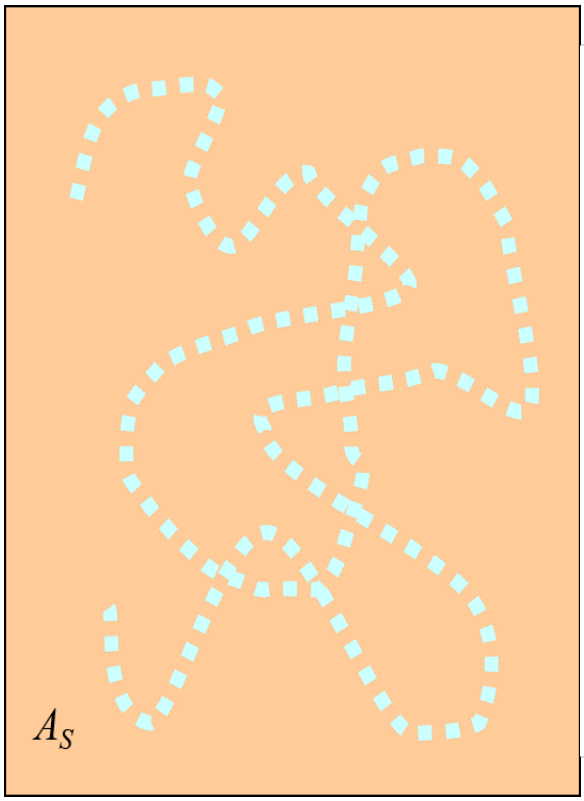


Figure 2.4. Visualized Random Search Path. Source: [1].

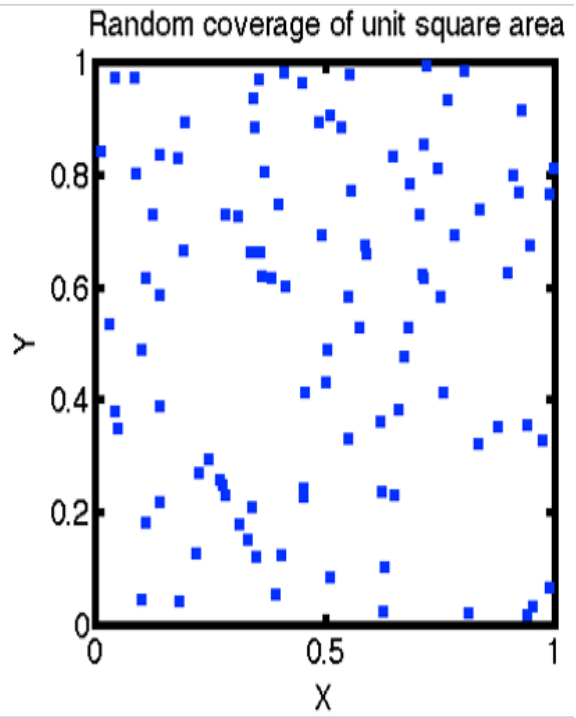


Figure 2.5. Modeled Search Path. Source: [1].

To determine the probability that a particular point is covered, consider the following:

- Consider

$$(v * w * t)/As = \gamma * t \tag{2.4}$$

- Each piece of confetti is size

$$(v * w * t)/n \tag{2.5}$$

- The probability that one piece covers a point is

$$((v * w * t)/n)/As = \gamma * t/n \tag{2.6}$$

- This means that

$$\gamma = v * w/As \tag{2.7}$$

- The probability that one piece misses point is

$$1 - \gamma * t/n \tag{2.8}$$

- The probability that all n pieces miss a point (by independence) is

$$(1 - \gamma * t/n)^n \tag{2.9}$$

- The probability that at least one piece covers point

$$1 - (1 - \gamma * t/n)^n \tag{2.10}$$

- As n approaches infinity recall:

$$\lim_{n\to\infty}(1 - \frac{x}{n})^n = e^{-x} \tag{2.11}$$

- So the probability a "searched confetti square" is covering a point is

$$1 - e^{-\gamma t} \tag{2.12}$$

- Therefore, the coverage as a function of time becomes

$$F(t) = 1 - e^{-\gamma t} \tag{2.13}$$

2.2 Model Overview

The following model, based on H. Wang and H. Zhou's research [7], incorporates Monte Carlo simulation to study the time evolution of the probability of detection of two searchers moving along a prescribed path, and a target undergoing Brownian diffusion. Utilizing the elements of an exhaustive search, an optimal search pattern will present itself based on varying speeds and search radii.

The model incorporates a single target and a set of two searchers. The target is undergoing Brownian motion with a given diffusion coefficient. Each of the two searchers move along a prescribed deterministic path, given by the code in Appendix C.

For each run, 100,000 independent Monte Carlo simulations are carried out simultaneously. At the beginning, an array of 100,000 entries (vectors) stores all initial positions of the target in 100,000 independent runs. Two other arrays of 100,000 entries (vectors) store the initial positions of the two searchers in 100,000 independent runs. The $j-th$ entry of target position array interacts with only the $j-th$ entry of searchers' position arrays. In this way, in each independent run, there is only one target and a set of two searchers.

The 100,000 independent runs are started with 100,000 targets uniformly distributed over the defined search region. This is to simulate the assumption that there is no preferred position for the target in the search region. During simulations, the target in each independent run moves randomly with a given diffusion coefficient and independently from targets in other runs.

Each of the two searchers moves according to a prescribed path, but other than this fact there is no real coordination or communication among the searchers. The separation between them for the in-line searches is input as a parameter before the simulation begins. In the multi-path ladder code, both targets begin at the same position, while one moves vertically and one moves horizontally (as seen in Figure 3.3). To simulate an in-line ladder search, an estimate had to be made as to allow searcher two to complete its horizontal cycle and begin at the initial position with a vertical search along the path searcher one was to take. This separation was estimated to be 4*Lx, where Lx is half the width of the search area.

In each independent run, if the target is within the detection range of any of the two searchers at any time, the target is detected, the event is recorded, and that particular run is terminated. This means that over time, the number of runs decreases. Over time, the detection probability is measured as the fraction of targets detected in the 100,000 independent runs.

CHAPTER 3:
Problem and Model

This chapter outlines the problem being investigated, and develops the model used to generate the conclusions.

3.1 The Problem

This model considers two surface ships prosecuting a submarine in the deep basin of the South China Sea. The searchers utilize three separate search paths, consisting of an in-line spiral search, an in-line ladder search, and a multi-path ladder search. These search paths are demonstrated in Figures 3.1 through 3.3.

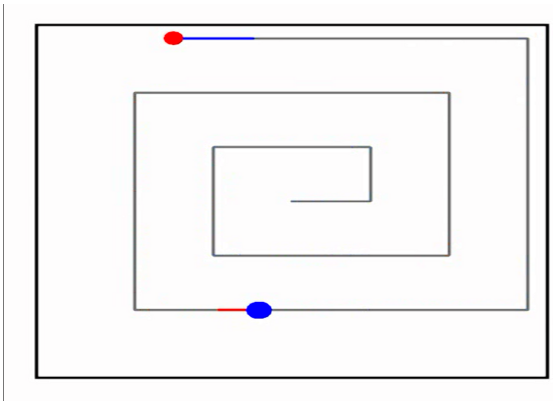


Figure 3.1. In-Line Spiral Search

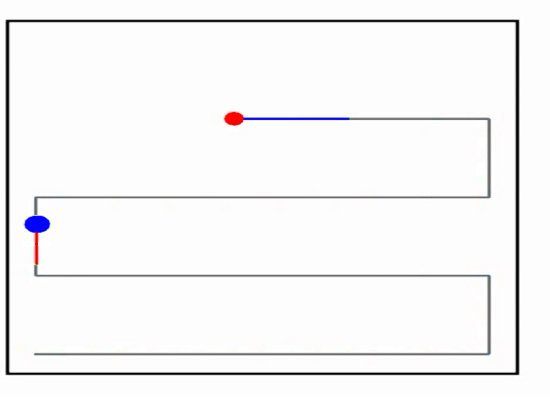


Figure 3.2. In-Line Ladder Search

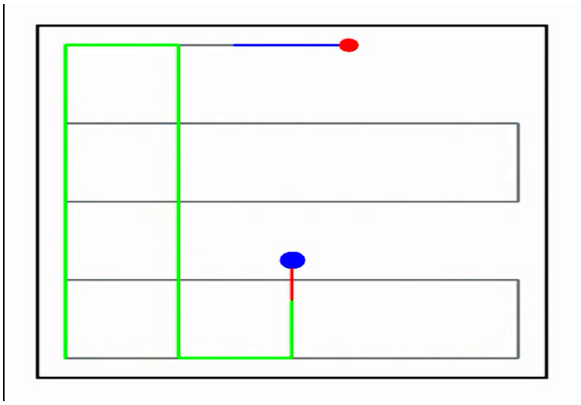


Figure 3.3. Multi-Path Ladder Search

Many varying factors impact the probability of detection, so the following parameters were varied during the course of the research.

Velocity Velocity impacts the depth of a towed sonar sensor, and in turn the detection range. Therefore to accurately estimate probability of detection, a range of velocities from 8 to 12 knots was utilized.

Sensor Range Since the range of detection relies heavily upon the physical attributes in the region (described later), a range of 1 to 3 nautical miles was applied based on climate models, and a range of 6 nautical miles was tested for experimentation purposes.

3.2 Geographic Information

The particular area being investigated is a central location in the South China Sea in the center of the deep basin in a 14nm by 14nm box centered at coordinates 21N 119E. The Google Earth image in Figure 3.4 shows the position relative to mainland China and Taiwan.

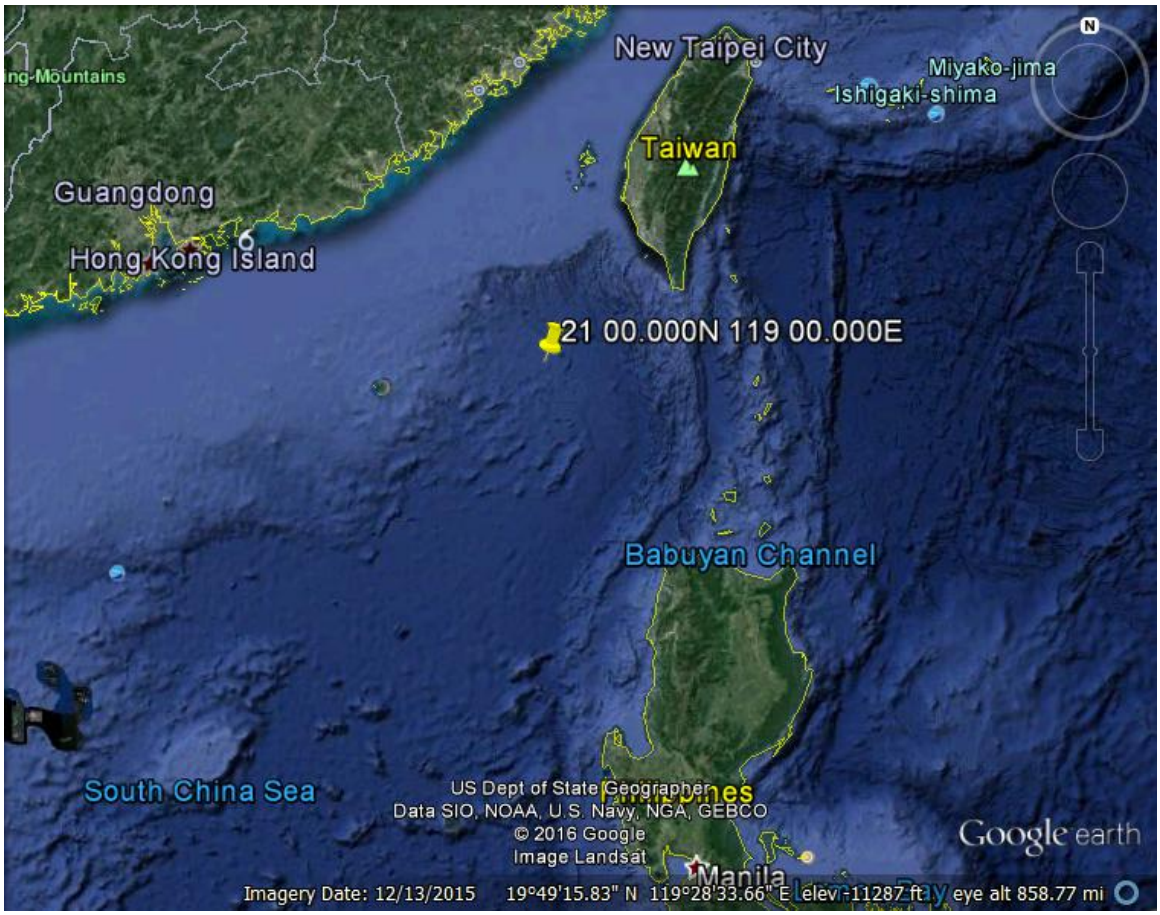


Figure 3.4. Search Area in the South China Sea
Image retrieved from Google Earth, https://www.google.com/earth/.

The Generalized Digital Environmental Model (GDEM) utilized by the Navy provides historical data regarding the temperature and salinity of virtually any part of the ocean. The ocean environment changes constantly, however the averaged data provided by GDEM provides information to obtain decently accurate models. Based on GDEM data for this particular area in the South China Sea, the depth in this box is approximately a constant 8,900 feet (relatively shallow compared to deeper ocean), and the bottom type is clay. These characteristics directly impact attenuation loss, described in the next section creating no convergence zones or bottom bounce, and requiring a direct path relationship to acquire the contact.

3.3 Environmental Factors/Physical Attributes

The speed of sound in water, and subsequently the range at which the sound can be detected, is directly proportional to three key factors: temperature, pressure, and salinity. An increase in any or all three factors creates an increase in sound speed, and conversely a decrease lowers the speed.

One of several models utilized by the Navy and researchers alike to estimate the ray paths and transmission loss experienced by different frequencies at varying depths in the ocean is called BELLHOP. In calculating the following profiles, a few constants were input into BELLHOP to determine the ray trace that a sound wave would follow, as well as the transmission loss and subsequently possible detection ranges that would occur. Those constants were assumed to be:

- Source Depth 164 ft (50m) and 328 ft (100m)
- Source Level 120 Hz

These constants were integrated with the following information to develop detection ranges.

3.3.1 Sound Speed Profile

To assist in determining the span of fixed ranges to utilize for this specific area in the South China Sea, the sound speed profiles (SSPs) for the designated area were collected from GDEM in the months of January and April, to represent cold and warm weather profiles. The profiles are shown below in Figure 3.5, along with a closer look at the differences near the surface layer in Figure 3.6.

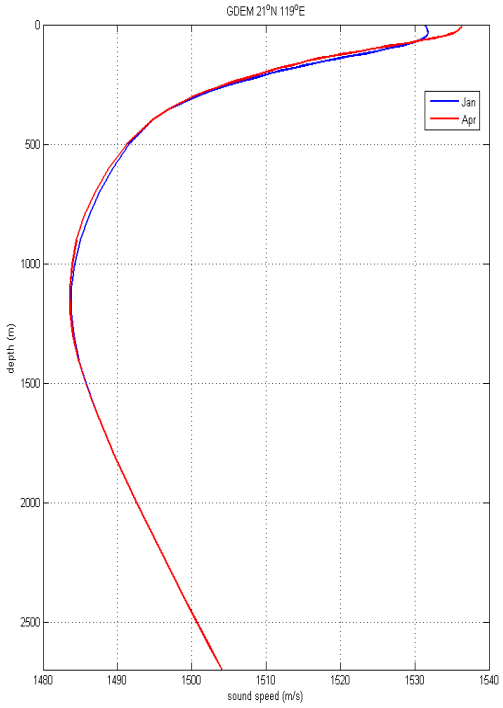


Figure 3.5. Sound Speed Profile

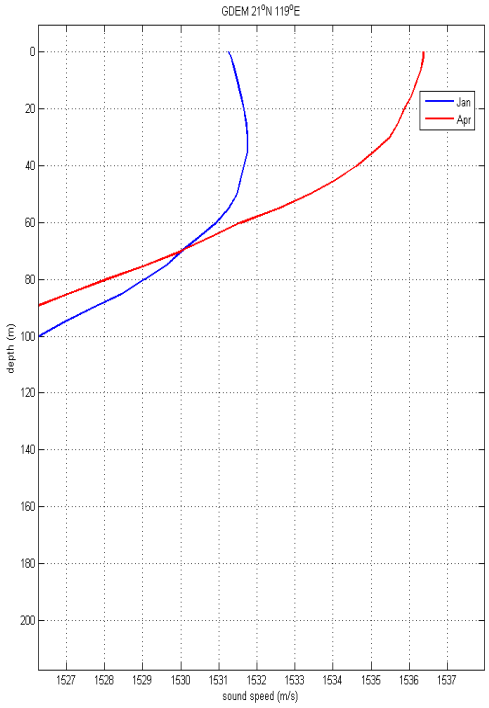


Figure 3.6. Sound Speed Profile at Surface

As seen in the figures, the colder months offer faster sound speeds near the surface due to surface layer warming by the sun (approximately 5 m/s or 11 mph), but as the water gets deeper (beyond 1,000 m or 3,280 ft) the sound speed becomes uniform.

3.3.2 Attenuation Loss/Detection Range

Employing the constants discussed earlier, and the associated SSPs, Figures 3.7 and 3.8 were obtained through BELLHOP regarding ray trace, which is the path that an unimpeded sound wave is modeled to follow regardless of frequency.

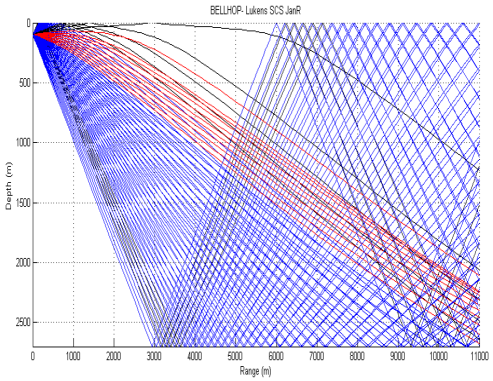


Figure 3.7. Jan Ray Trace

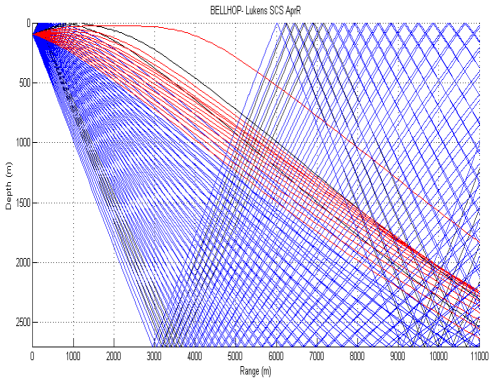


Figure 3.8. Apr Ray Trace

Note that with the source at 328 ft (100m), the ray trace in Figures 3.7 and 3.8 shows that a sensor at 1,200 ft (335m) should detect direct path sound out to 3.77 nm (7,000m) in January and 2.7 nm (5,000m) in April, and sound from a bottom bounce path out to almost 6 nm (11,000m). However, the ray trace model does not take energy loss due to bottom type or spherical spreading into account, and transmission loss can show a much different picture.

Figures 3.9 through 3.12 are products of the BELLHOP model estimating transmission loss (TL). As previously discussed, the ray paths in Figures 3.7 and 3.8 do not account for TL and the path a ray will realistically follow. Figures 3.9 through 3.12 display the TL and subsequent ranges, based on the possible 1200 ft (335m) depth of an AN/SQR-19 Tactical Towed Array Sonar (TACTAS) [9].

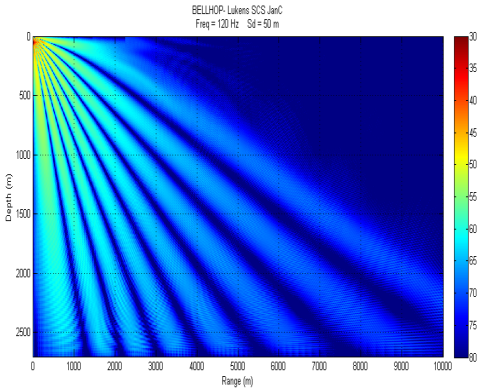


Figure 3.9. Jan TL, Source at 50m

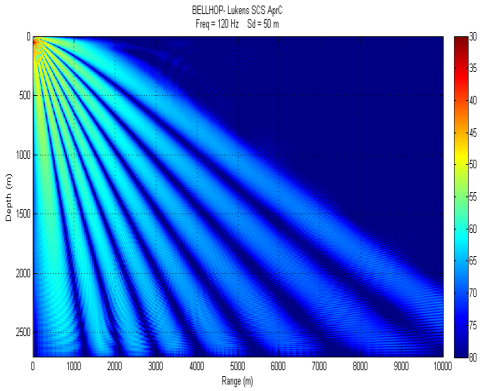


Figure 3.10. Apr TL, Source at 50m

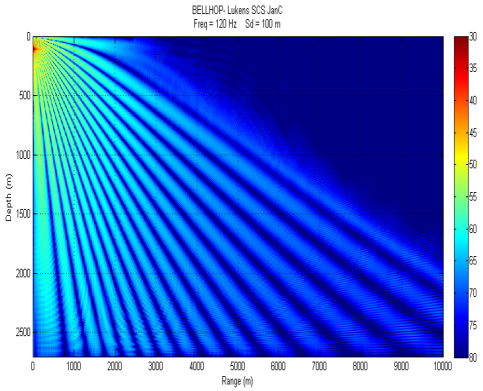


Figure 3.11. Jan TL, Source at 100m

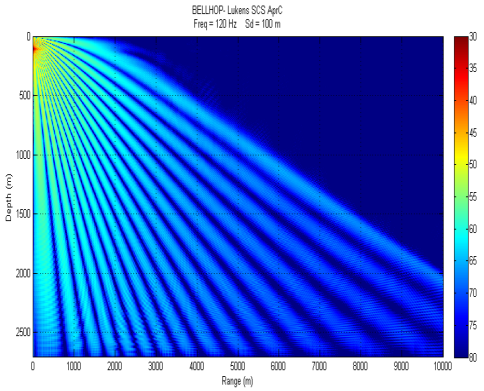


Figure 3.12. Apr TL, Source at 100m

As seen in the transmission loss, Figures 3.9 through 3.12, the bottom bounce has disappeared and direct path contact only exists out to a maximum of 2.1 nm (4,000m) with a sensor at 1,200 ft (335m). As a result, search radii from a range of .5 to 2.5 nm were utilized for the searchers.

Several other assumptions built into the model include a total time of 24 hours spent searching for the target, as well as allowing the searchers to occupy the same point at the same time. In a real world scenario, this would involve the collision of two surface vessels, but for coding purposes this fact is overlooked for simplicity's sake.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results

The following chapter outlines the results yielded from the three separate search patterns. Recall that the two primary methods investigated for each scenario were to hold the searchers' velocities constant and vary the search radii, followed by holding the search radii constant and varying the velocities.

Since there are a total of 24 tables and 360 graphs, the following analysis will focus on a small sample of the results to show general trends that were observed. Many more results appear in Appendices A and B.

4.1 In-Line Ladder Search

4.1.1 Time to 90 Percent Probability of Detection

Table 4.1 shows the time (in hours) until a 90 percent probability of detection was reached for the in-line ladder search. As the results were very similar between the two research methods, this table represents holding velocities constant at 10 kts and varying the search radii.

Table 4.1. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=12.28	T=9.94	T=7.63	T=4.94	T=4.28
	1	T=8.97	T=5.88	T=5.00	T=4.50	T=3.75
Ship 1	1.5	T=5.66	T=5.00	T=4.78	T=4.50	T=3.66
	2	T=5.16	T=4.88	T=4.78	T=4.50	T=4.00
	2.5	T=4.97	T=4.75	T=4.50	T=4.50	T=3.91

For the in-line ladder search the 2.5 nm search radii for both searchers naturally provided the fastest time to 90 percent probability of detection at 3.91 hours. An interesting observation is how little the time changes as the search radii increased. Note that at the smallest value of .5 nm for searchers one and two, the time is 12.28 hours. Increasing the radius to 1 nm for

one of the searchers cuts the time by 20-40%, while increasing both of the searchers' radii decreases the time by nearly 50%. This is the largest observable jump in the data. However, as the radii increases beyond this point, the change in time is much less drastic. This trend is also apparent in the other tables for the in-line ladder search found in Appendix B. With both velocity and search radius at a maximum for both searchers, the in-line ladder search offered the fastest time of the three search patterns.

4.1.2 Radius/Velocity Investigation

The results drawn from the simulations when holding velocity constant and varying the search radius proved to be quite similar to those seen by holding search radius constant and varying velocity. Therefore, for the sake of comparison and demonstrating trends, Figures 4.1 through 4.6 provide the results from both searchers having the same search radius and the same velocities. The figures presented represent the slowest and fastest velocities investigated (8 and 10 kts), and the search radii of .5, 1.5, and 2.5 nm.

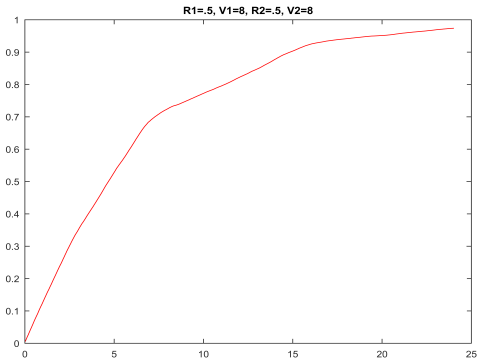


Figure 4.1. R=.5, V=8

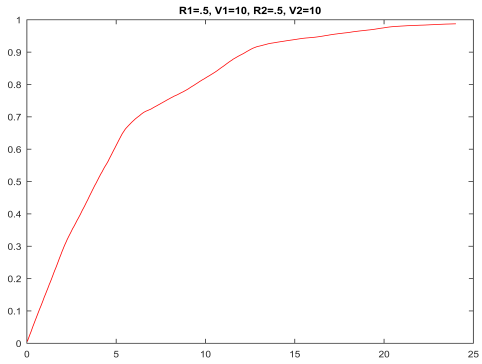


Figure 4.2. R=.5, V=10

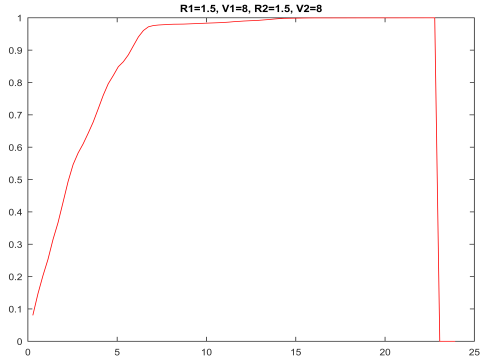


Figure 4.3. R=1.5, V=8

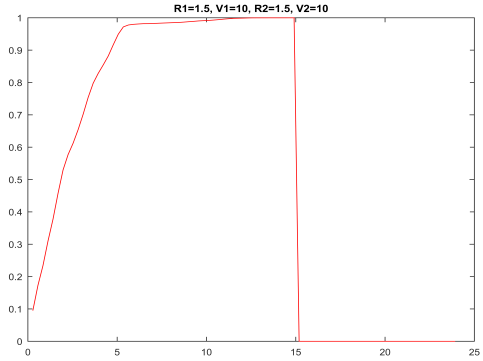


Figure 4.4. R=1.5, V=10

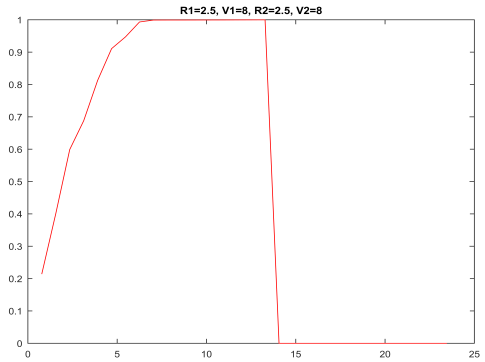


Figure 4.5. R=2.5, V=8

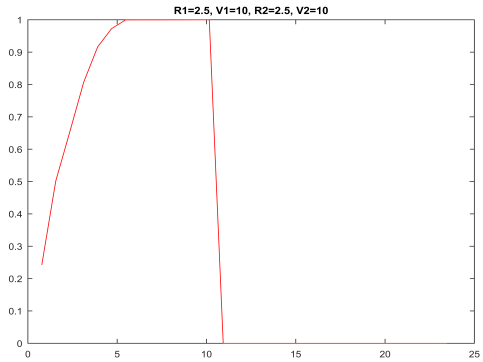


Figure 4.6. R=2.5, V=10

As seen in the figures, increasing speed or search radius size leads to a higher probability of detection being reached at a faster rate. The point at which the simulation reached an overall probability of detection of 1, the simulation then stops which causes the vertical line in the graph. This means with the in-line ladder search, with a maximum radius and speed for both searchers the contact was guaranteed to be found in approximately 11 hours, but even with a radius of 1.5 nm the probability of detection was 1 within the 24 hour time period of searching. The .5 nm search radius values began to approach 1 at the end of the search period, but did not reach it completely.

4.2 Multi-Path Ladder Search

4.2.1 Time to 90 Percent Probability of Detection

Table 4.2 shows the time (in hours) until a 90 percent probability of detection was reached for the multi-path ladder search, again holding a constant 10 kts and varying the search radii.

Table 4.2. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=13.00	T=8.75	T=7.12	T=6.88	T=6.69
	1	T=8.38	T=7.25	T=6.88	T=6.63	T=6.63
Ship 1	1.5	T=6.88	T=6.63	T=6.75	T=6.47	T=6.47
	2	T=6.06	T=5.75	T=5.63	T=6.00	T=5.50
	2.5	T=5.44	T=5.34	T=5.34	T=5.50	T=5.47

The multi-path ladder search fastest time was 5.47 hours to 90% probability of detection, with the search radii at their largest value of 2.5 nm for both searchers. This is very similar to the in-line ladder search, with the exception that it is slower. In fact, it is the slowest time of the three separate search patterns. The same large jump is also observed if one or both searchers increases its search radius beyond .5 nm, but anything larger than that has much less of an impact.

4.2.2 Radius/Velocity Investigation

Figures 4.7 through 4.12 show the results from the multi-path ladder search, and the format follows the same parameters listed for the in-line ladder search.

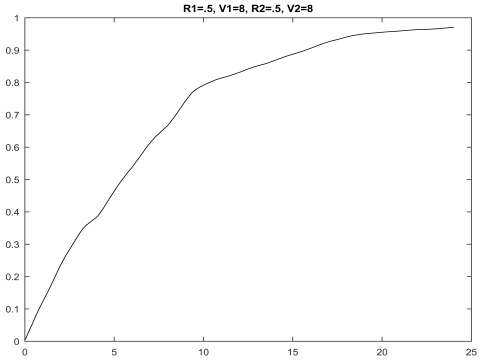


Figure 4.7. R=.5, V=8

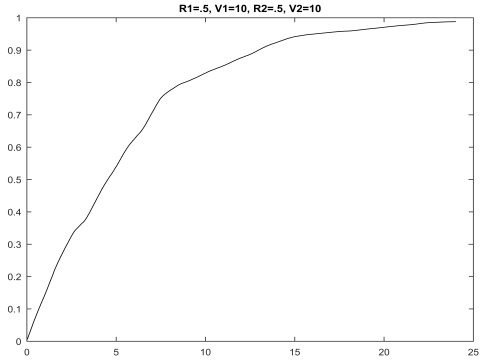


Figure 4.8. R=.5, V=10

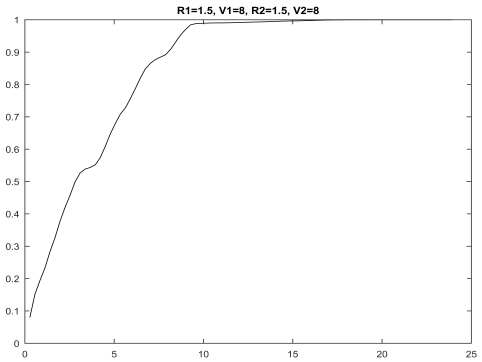


Figure 4.9. R=1.5, V=8

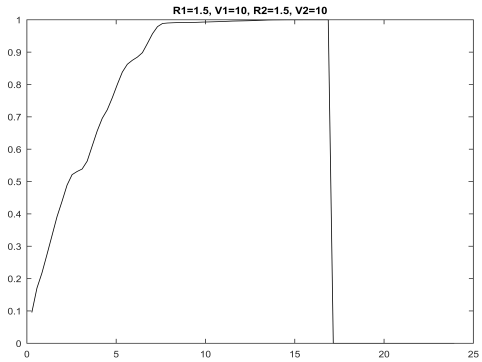


Figure 4.10. R=1.5, V=10

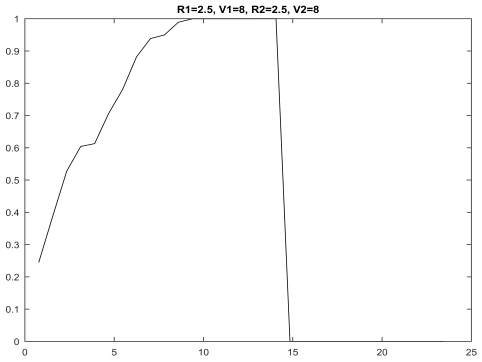


Figure 4.11. R=2.5, V=8

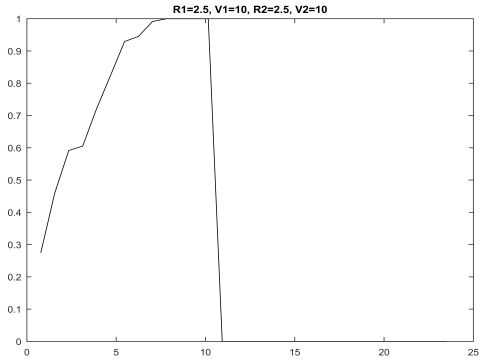


Figure 4.12. R=2.5, V=10

Similar to the in-line ladder, when the searchers' radii is .5 nm in the multi-path ladder search, increasing speed has little impact on the rate at which probability of detection increases.

Also similar to the in-line search, with the searchers' parameters at maximum velocities and search radii, the multi-path search also takes approximately 11 hrs to guarantee detection. However, unlike the in-line search, the multi-path does not reach a probability of detection of 1 until the velocity is increased to 10 kts with the search radii at 1.5 nm.

4.3 In-Line Spiral Search

4.3.1 Time to 90 Percent Probability of Detection

Table 4.3 shows the time (in hours) until a 90 percent probability of detection was reached for the multi-path ladder search, again holding a constant 10 kts and varying the search radii.

Table 4.3. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=17.75	T=9.88	T=7.38	T=6.13	T=4.25
	1	T=10.34	T=9.00	T=7.50	T=6.00	T=4.50
Ship 1	1.5	T=8.50	T=7.50	T=6.75	T=6.75	T=4.50
	2	T=7.25	T=7.00	T=6.75	T=6.00	T=6.00
	2.5	T=6.00	T=6.00	T=5.63	T=5.00	T=4.69

The fastest time to 90% probability of detection for the in-line spiral search was 4.69 hours, making it the second-fastest time of the three search patterns. Similar to the two ladder searches, this occurred at the maximum values of velocity and search radius. The same large jump is observed increasing the search radius from .5 nm to any other value, but again the other changes are rather miniscule in comparison.

4.3.2 Radius/Velocity Investigation

Figures 4.13 through 4.18 show the results from the in-line spirla search, and the format follows the same parameters listed for the ladder searches.

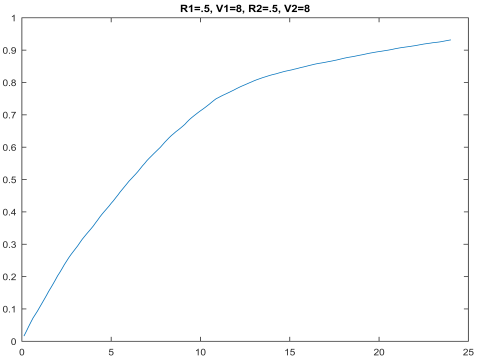


Figure 4.13. R=.5, V=8

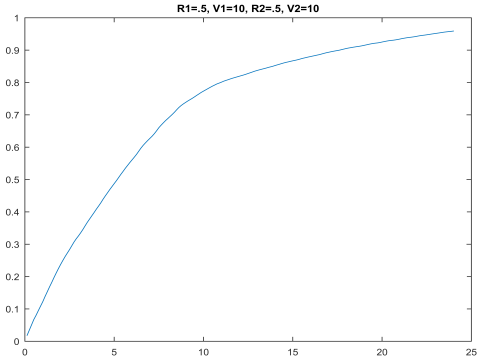


Figure 4.14. R=.5, V=10

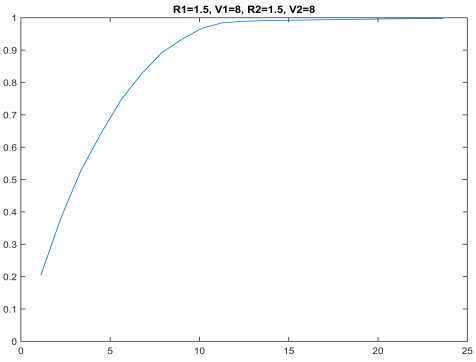


Figure 4.15. R=1.5, V=8

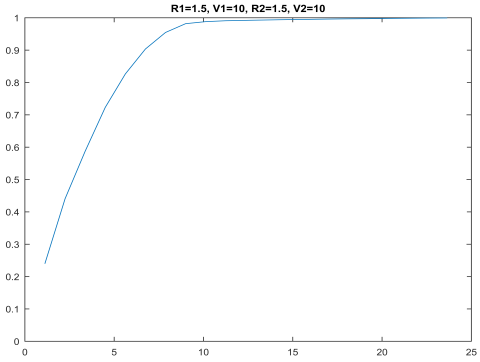


Figure 4.16. R=1.5, V=10

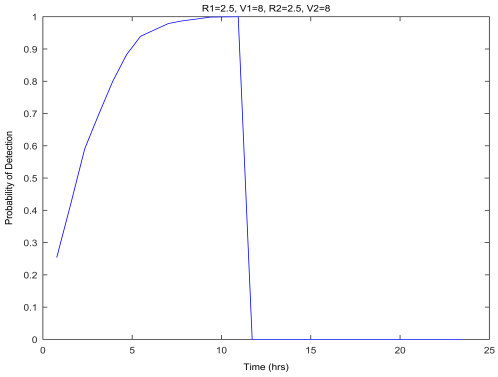


Figure 4.17. R=2.5, V=8

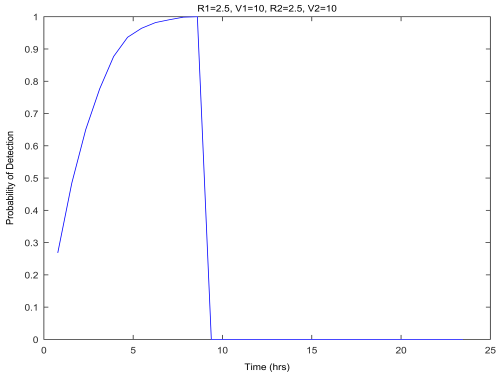


Figure 4.18. R=2.5, V=10

As seen in the figures, the in-line spiral search depends more on larger search radii for the searchers than a higher velocity. The results for .5 nm at 8 and 10 kts are quite similar,

and the same is true for a 1.5 nm radius. However, while a probability of detection of 1 is not reached until 2.5 nm, the time taken to reach that point is faster than the ladder search results. At maximum parameter values, the spiral search only takes approximately 9 hours to reach guaranteed location of the target. Even at the slower value of 8 kts, it takes approximately 12 hours, which is only 1 hour slower than the fastest ladder search times.

CHAPTER 5:
Conclusions and Future Work

In summary, an in-line spiral search, in-line ladder search, and multi-path ladder search provided the search patterns to explore in this research. The results of the probability of detection are as follows.

5.1 Results

Based on the formulas for an exhaustive search outlined in Chapter 2, the results from running 100,000 Monte Carlo simulations in all three search patterns yielded the following intuitive results:

- As speed increases, probability of detection increases.
- As search radius increases, probability of detection increases.
- The results from fixing the search radius and varying speed of the searchers are very similar to fixing the velocity and varying the search radius.

The results above are easily observed in the simulation results graphed in Appendix 1. As either the search radius R or the velocity V were varied as the other parameter was held constant, the time to complete (or nearly complete) coverage decreased greatly. When either of the two factors were large enough, the results converged to a probability of detection of 1, and this fact is noted by a vertical line terminating the graph.

The simulations also yielded results that were not as intuitive:

- With a larger search radius, the in-line spiral search provides a faster time to full coverage.
- The in-line ladder and multi-path ladder converge at a smaller radii than the in-line spiral.
- The results between the in-line ladder and multi-path ladder are relatively similar.

At the maximum values of R and V for the in-line spiral search, the time to full coverage occurs at approximately 9 hours, as seen in Appendix A, where the ladder searches both

converge at approximately 11 hours. However, the two ladder searches begin to converge for much smaller values of R and V than the spiral search.

5.2 Future Work and Conclusions

Many more applications of this research to future work still exist. Varying all possible combinations of R and V alone provides 2,160 possible combinations. This work focused on only 360 of those combinations due to time constraints. Furthermore, a search radius that changes as a function of velocity or time provides a more realistic output in an ASW scenario, as velocity affects the depth of the towed array and in turn the detection range. Finally, the size of the search area can be increased, or moved to a completely different locations. Similar climatological research must be completed to determine new detection ranges, however the process will remain the same.

In conclusion, while the in-line spiral search offers a faster time to complete coverage of a search area, the ladder searchers provide complete coverage for smaller values of R and V. Therefore, the in-line ladder or multi-path ladder searches are recommended as the primary search method for two assets prosecuting a single contact in the South China Sea.

APPENDIX A:
Simulation Figures

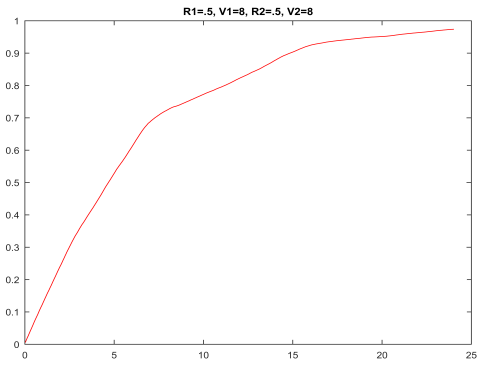
The following sections outline the final probability of detection based on various ranges of search speeds and search radii for the two searchers. All figures are presented with probability of detection on the y-axis as a function of time on the x-axis.

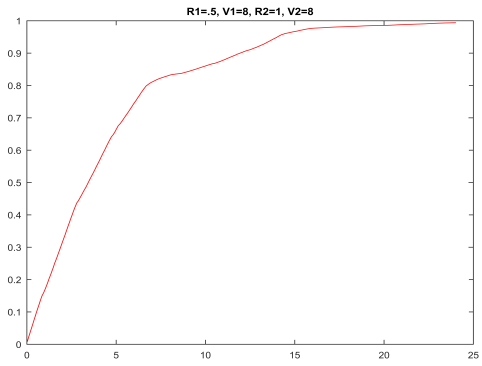
A.1 In-Line Ladder Search

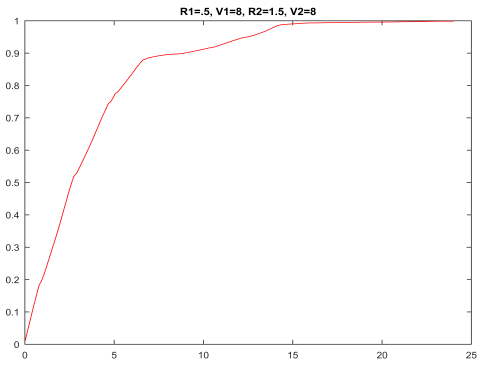
The following figures are for two searchers conducting an in-line ladder search.

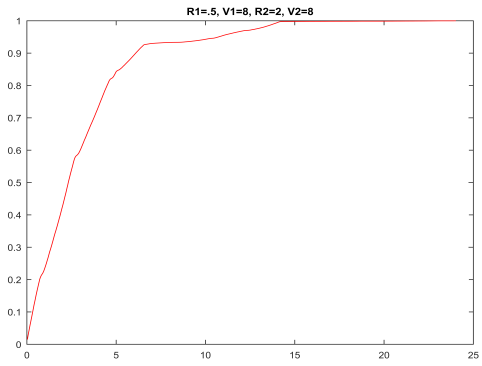
A.1.1 Varying Search Radii, Constant Speed

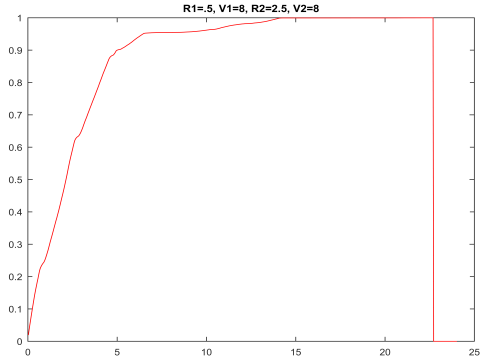
8 kts

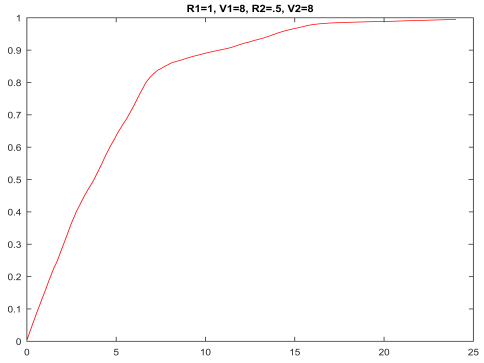


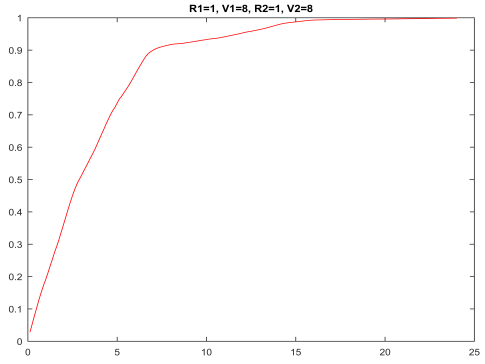


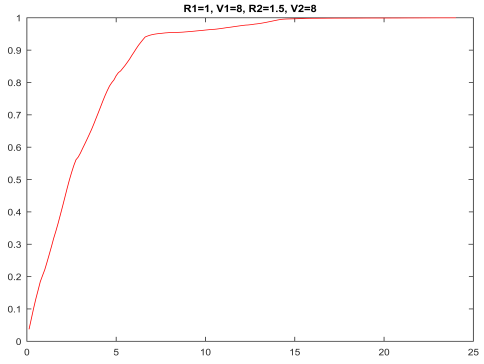


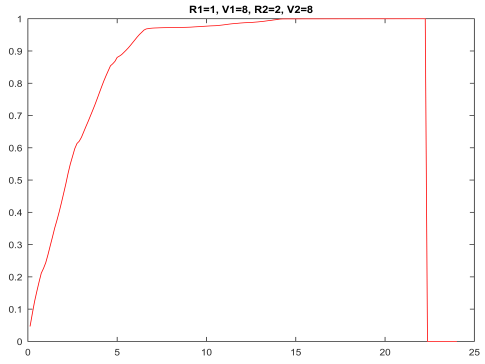


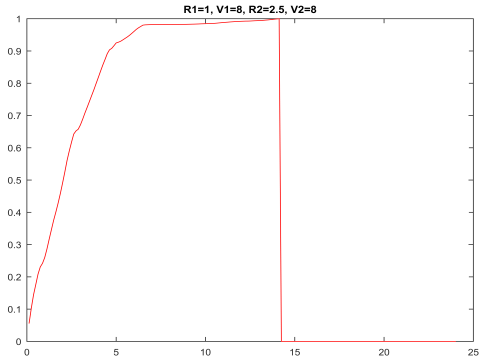

R1=.5, V1=8, R2=2.5, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

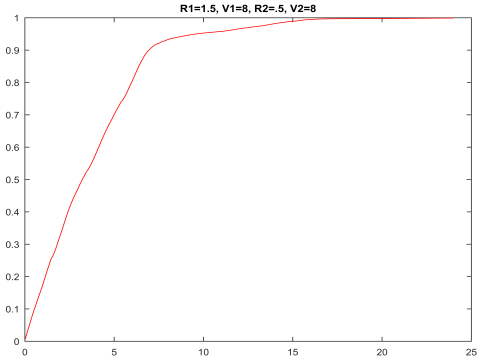

R1=1, V1=8, R2=.5, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

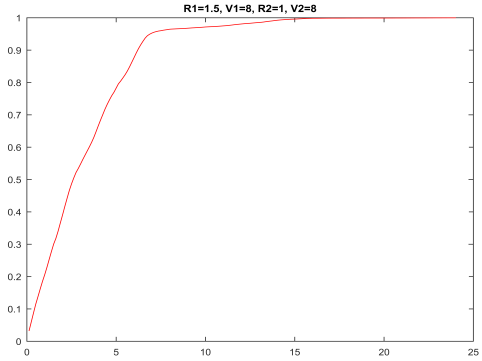

R1=1, V1=8, R2=1, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

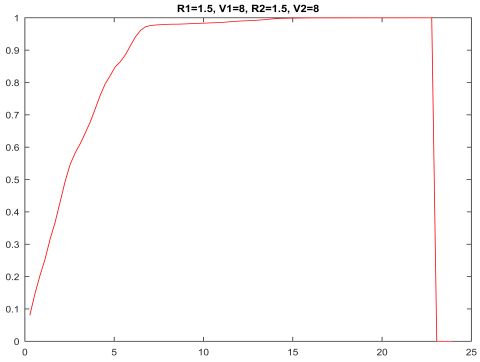

R1=1, V1=8, R2=1.5, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

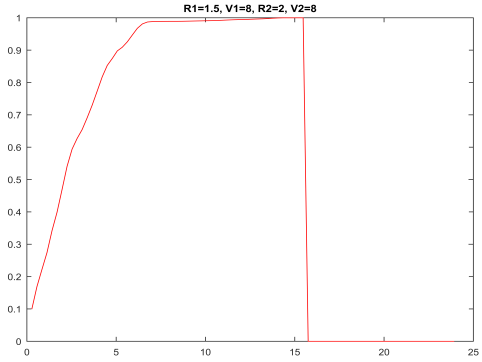

R1=1, V1=8, R2=2, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

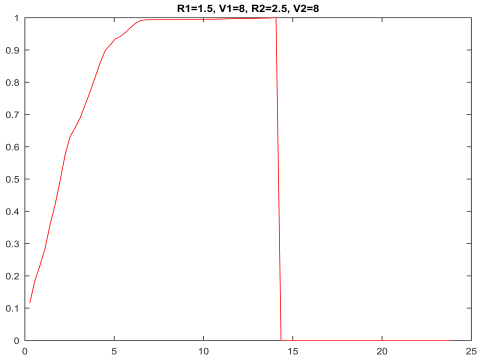

R1=1, V1=8, R2=2.5, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

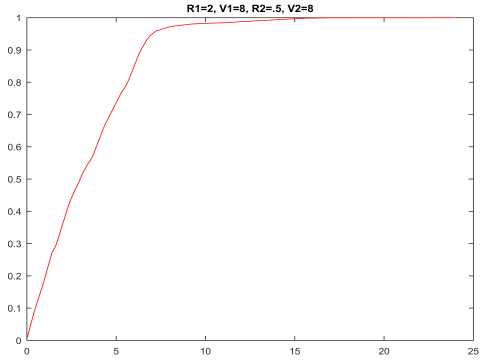

R1=1.5, V1=8, R2=.5, V2=8

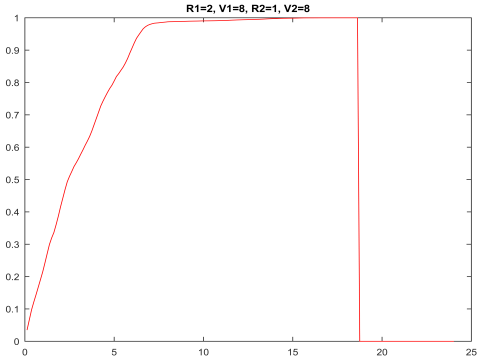

R1=1.5, V1=8, R2=1, V2=8

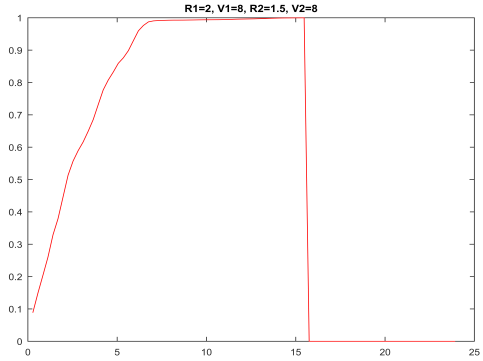

R1=1.5, V1=8, R2=1.5, V2=8

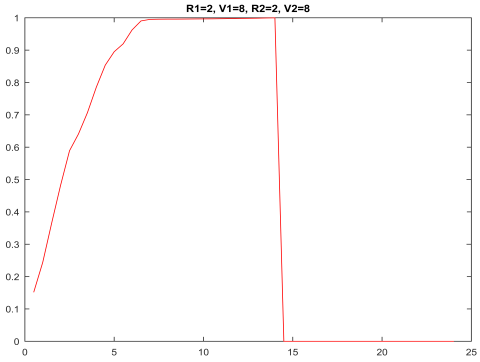

R1=1.5, V1=8, R2=2, V2=8

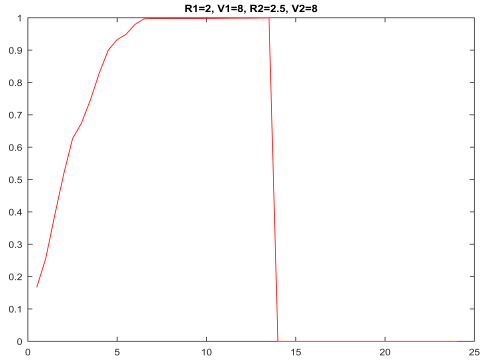

R1=1.5, V1=8, R2=2.5, V2=8

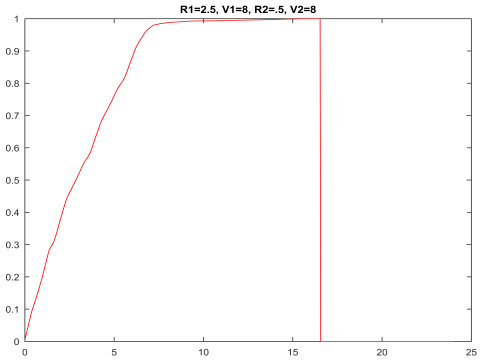

R1=2, V1=8, R2=.5, V2=8

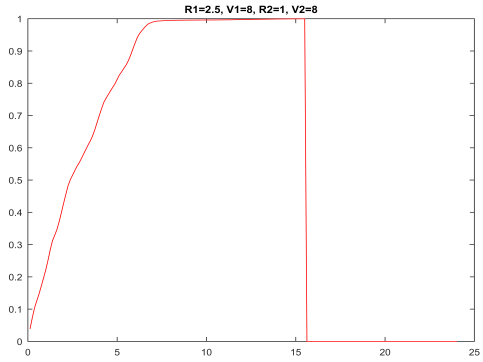

R1=2, V1=8, R2=1, V2=8

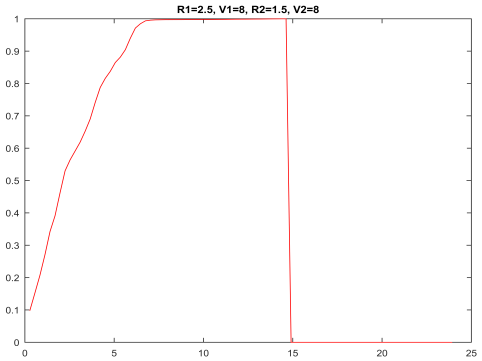

R1=2, V1=8, R2=1.5, V2=8

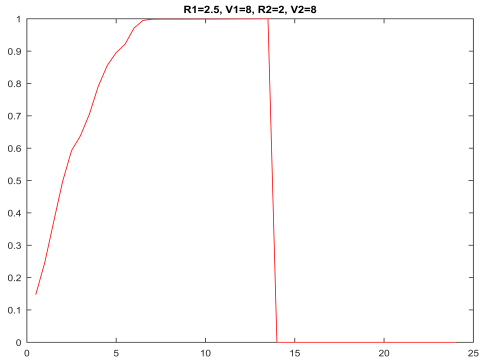

R1=2, V1=8, R2=2, V2=8

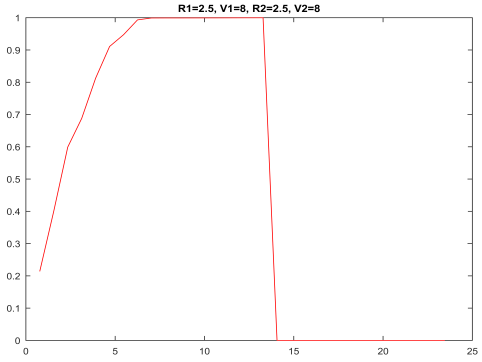

R1=2, V1=8, R2=2.5, V2=8


R1=2.5, V1=8, R2=.5, V2=8

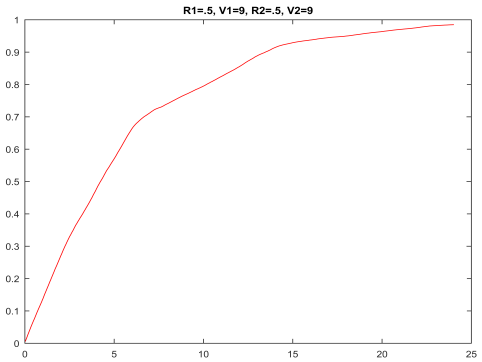

R1=2.5, V1=8, R2=1, V2=8

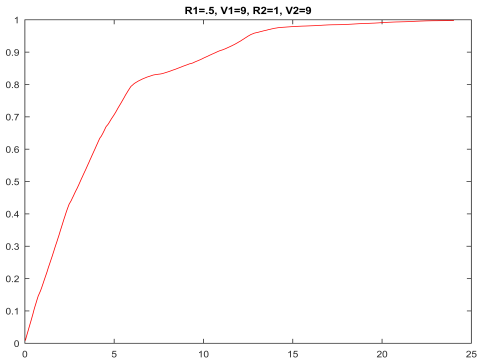

R1=2.5, V1=8, R2=1.5, V2=8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

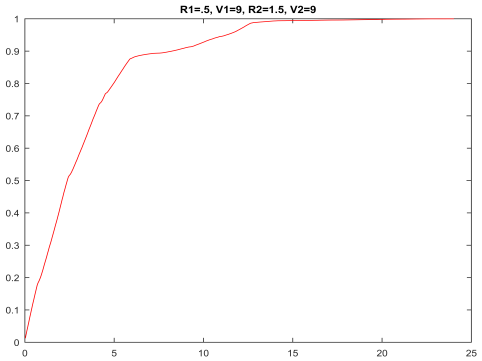

R1=2.5, V1=8, R2=2, V2=8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

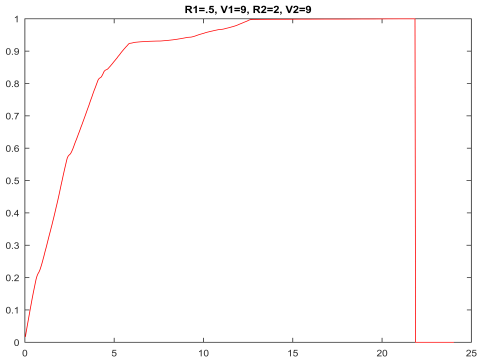

R1=2.5, V1=8, R2=2.5, V2=8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

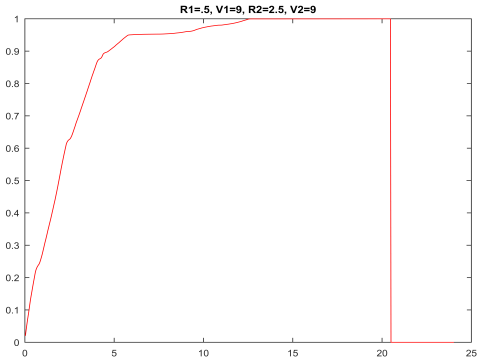
9 kts

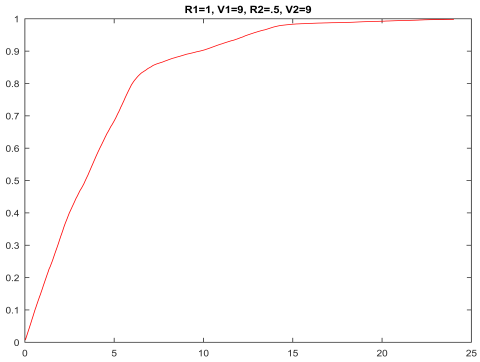

R1=.5, V1=9, R2=.5, V2=9

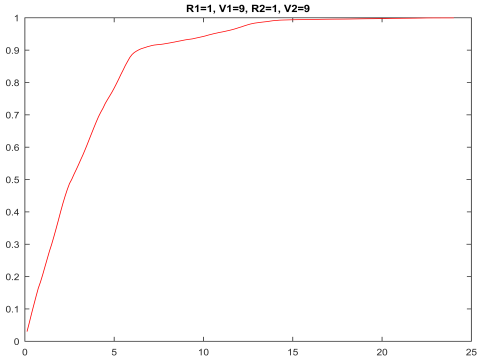

R1=.5, V1=9, R2=1, V2=9

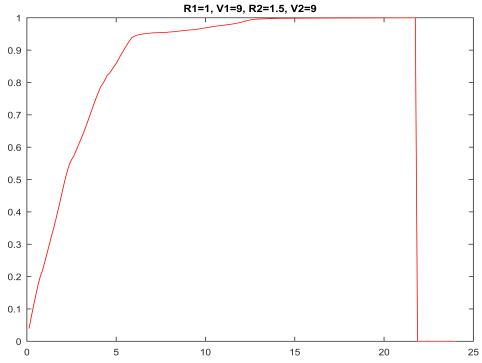

R1=.5, V1=9, R2=1.5, V2=9

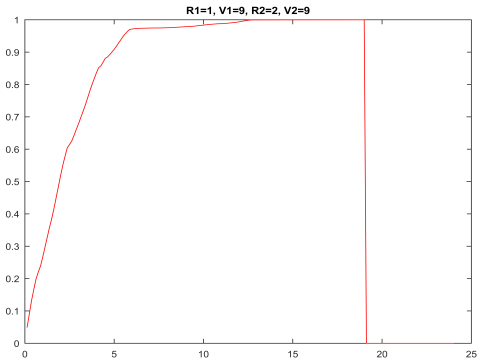

R1=.5, V1=9, R2=2, V2=9

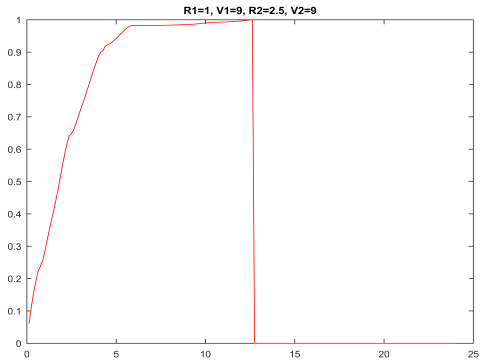

R1=.5, V1=9, R2=2.5, V2=9

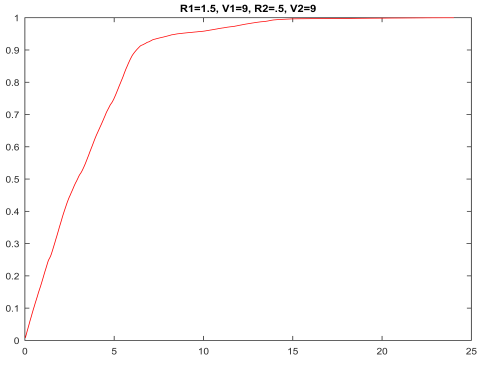

R1=1, V1=9, R2=.5, V2=9

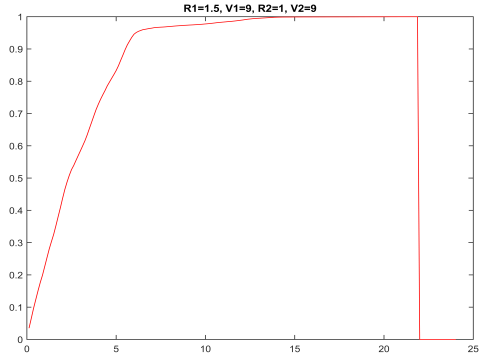

R1=1, V1=9, R2=1, V2=9

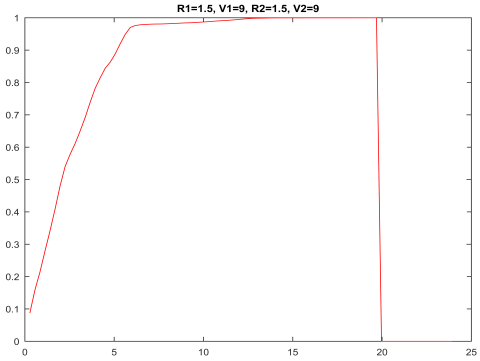

R1=1, V1=9, R2=1.5, V2=9

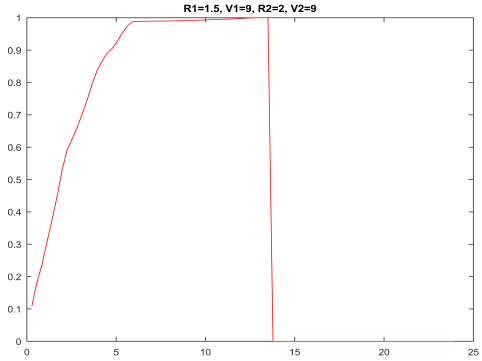

R1=1, V1=9, R2=2, V2=9

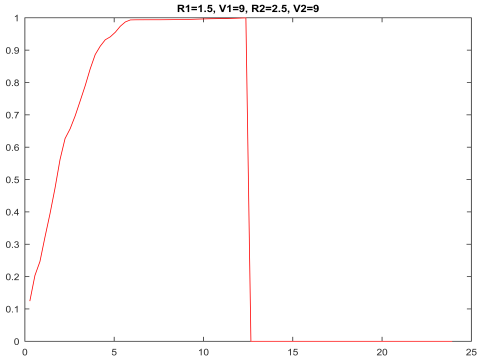

R1=1, V1=9, R2=2.5, V2=9

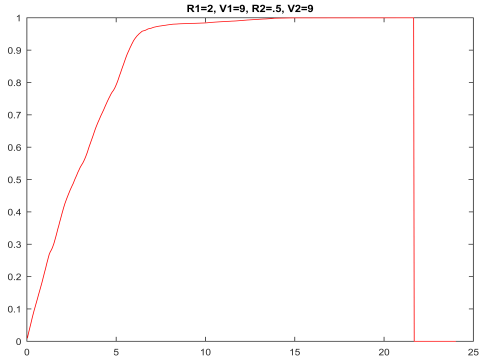

R1=1.5, V1=9, R2=.5, V2=9

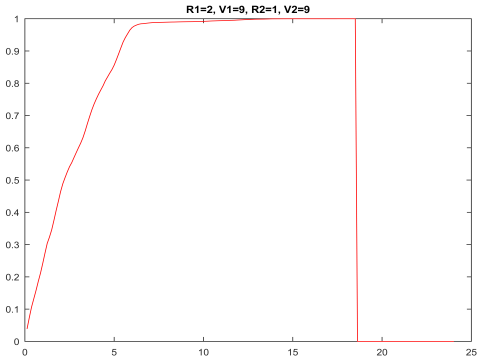

R1=1.5, V1=9, R2=1, V2=9

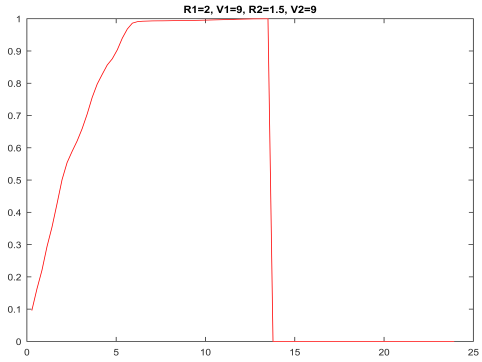

R1=1.5, V1=9, R2=1.5, V2=9

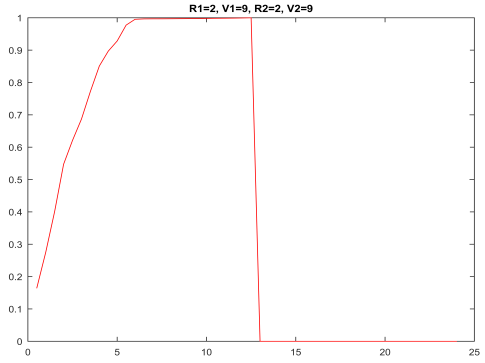

R1=1.5, V1=9, R2=2, V2=9

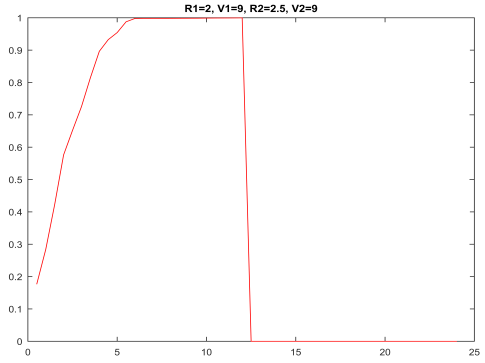

R1=1.5, V1=9, R2=2.5, V2=9

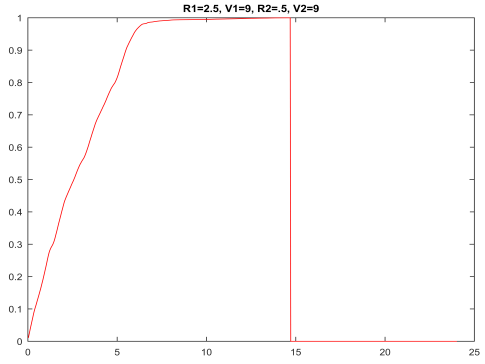

R1=2, V1=9, R2=.5, V2=9

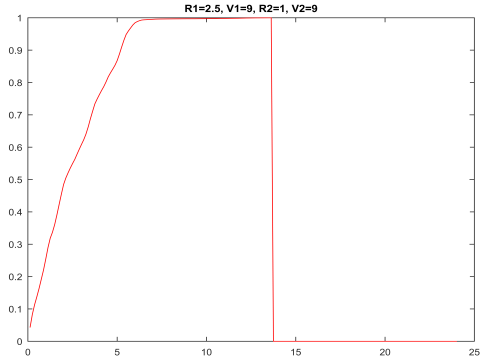

R1=2, V1=9, R2=1, V2=9

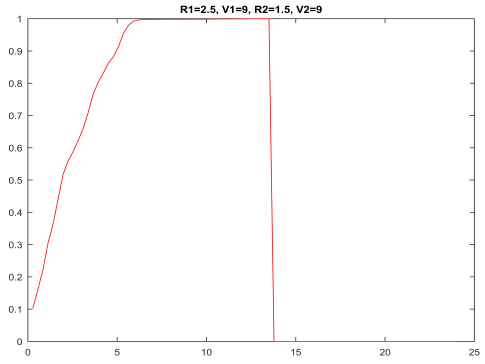

R1=2, V1=9, R2=1.5, V2=9

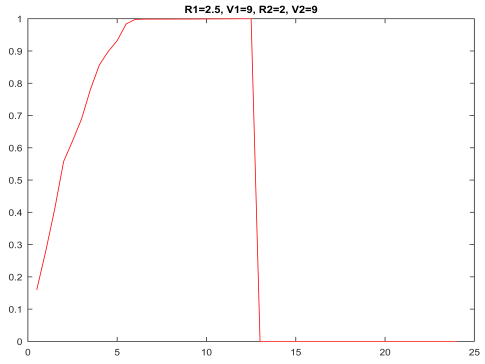

R1=2, V1=9, R2=2, V2=9

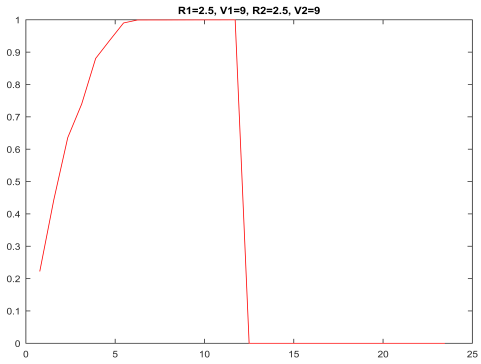

R1=2, V1=9, R2=2.5, V2=9


R1=2.5, V1=9, R2=.5, V2=9

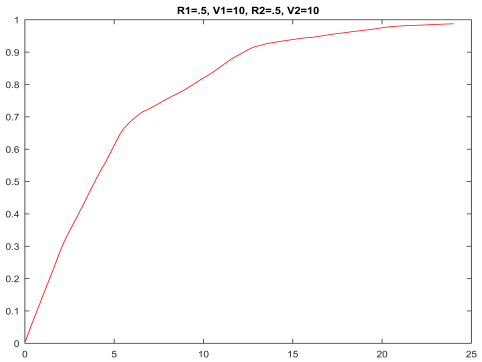

R1=2.5, V1=9, R2=1, V2=9

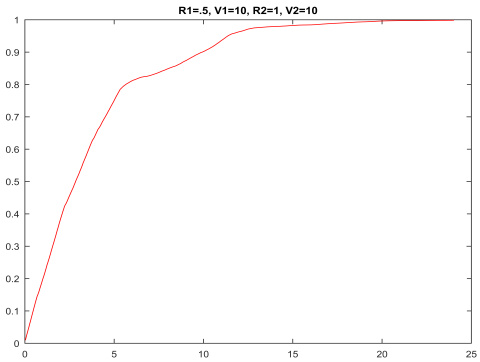

R1=2.5, V1=9, R2=1.5, V2=9

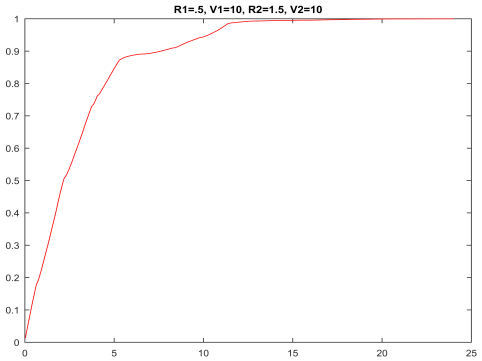

R1=2.5, V1=9, R2=2, V2=9

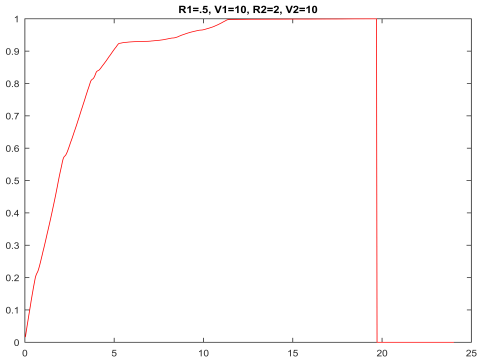

R1=2.5, V1=9, R2=2.5, V2=9
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

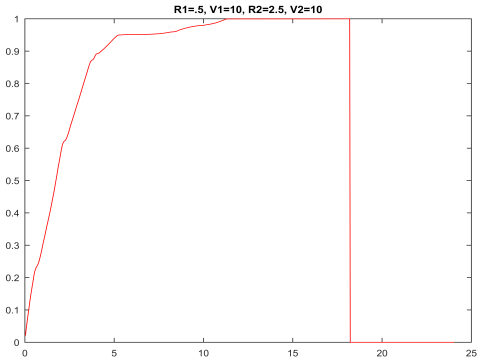
10 kts

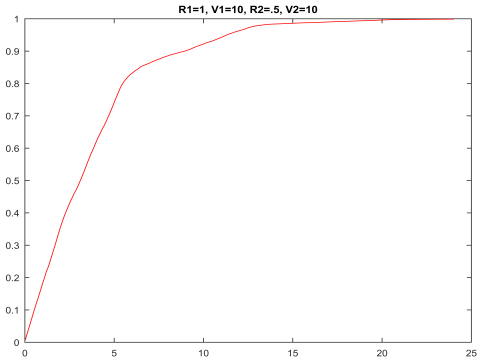

R1=.5, V1=10, R2=.5, V2=10

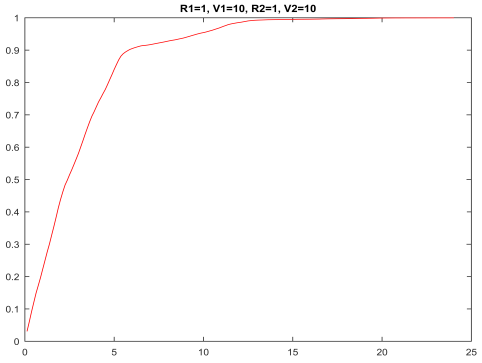

R1=.5, V1=10, R2=1, V2=10

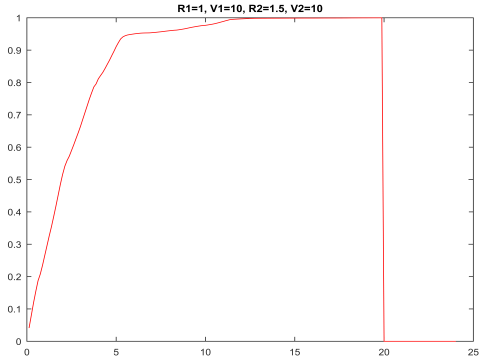

R1=.5, V1=10, R2=1.5, V2=10

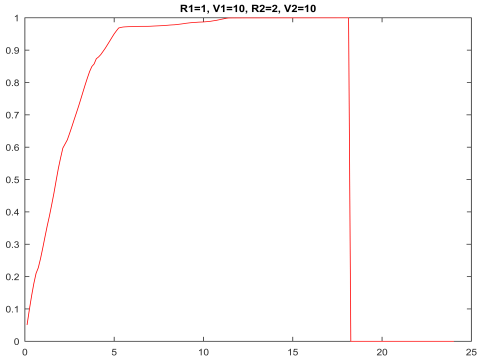

R1=.5, V1=10, R2=2, V2=10

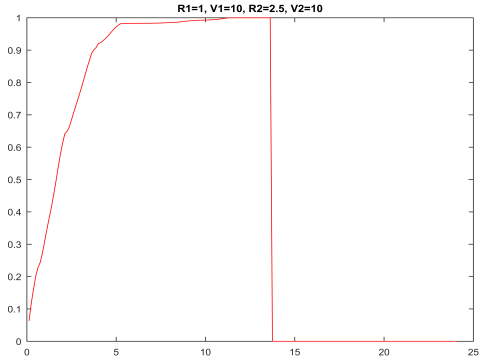

R1=.5, V1=10, R2=2.5, V2=10

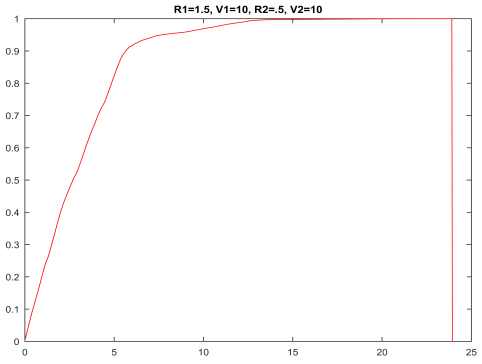

R1=1, V1=10, R2=.5, V2=10

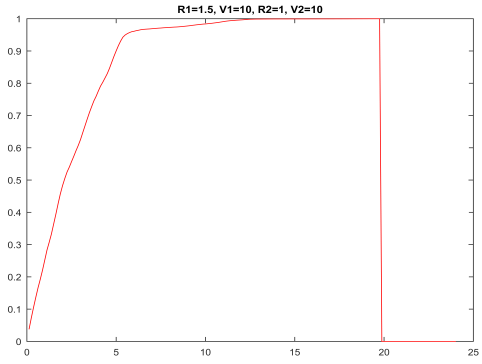

R1=1, V1=10, R2=1, V2=10

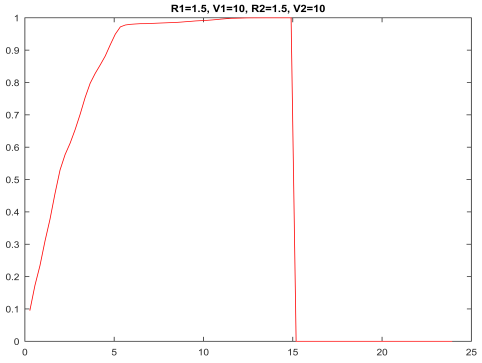

R1=1, V1=10, R2=1.5, V2=10

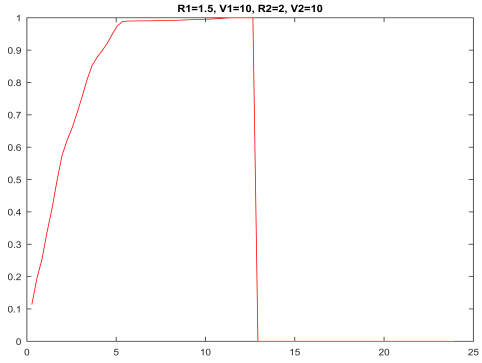

R1=1, V1=10, R2=2, V2=10

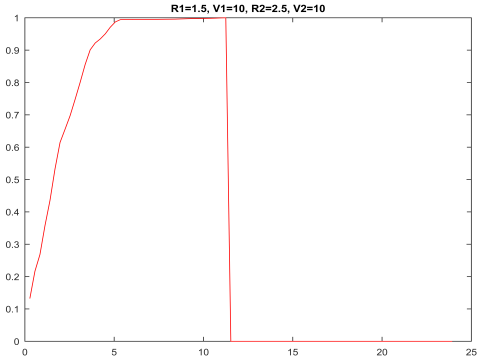

R1=1, V1=10, R2=2.5, V2=10

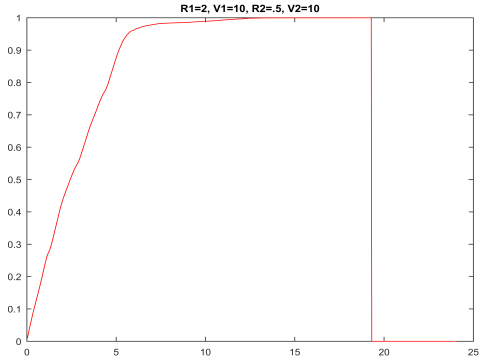

R1=1.5, V1=10, R2=.5, V2=10

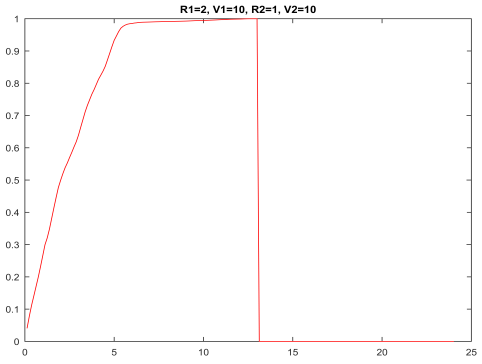

R1=1.5, V1=10, R2=1, V2=10

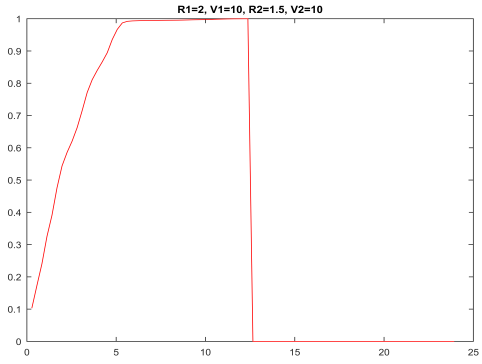

R1=1.5, V1=10, R2=1.5, V2=10

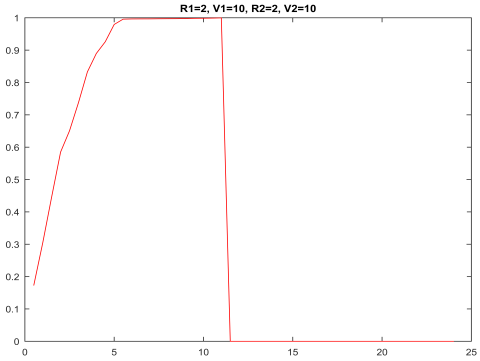

R1=1.5, V1=10, R2=2, V2=10

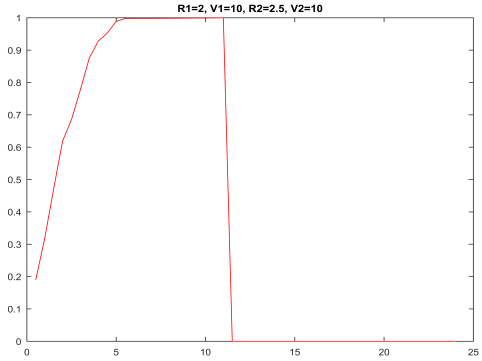

R1=1.5, V1=10, R2=2.5, V2=10

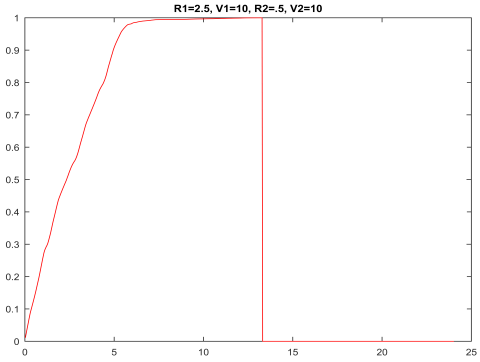

R1=2, V1=10, R2=.5, V2=10

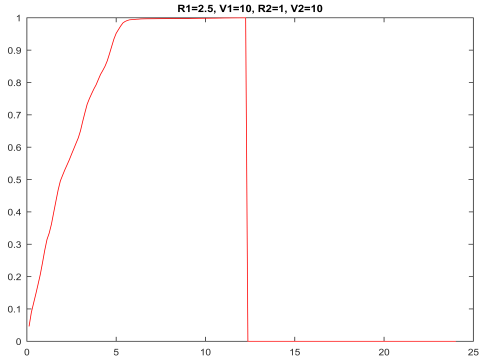

R1=2, V1=10, R2=1, V2=10

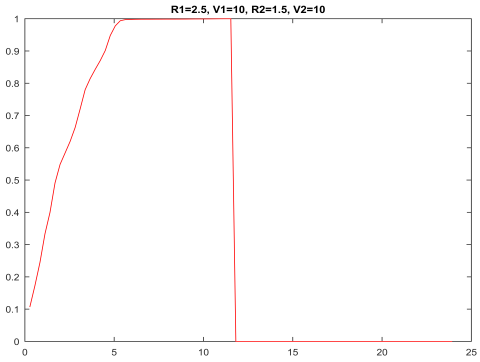

R1=2, V1=10, R2=1.5, V2=10

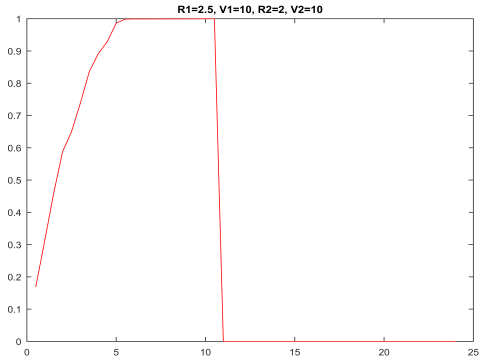

R1=2, V1=10, R2=2, V2=10

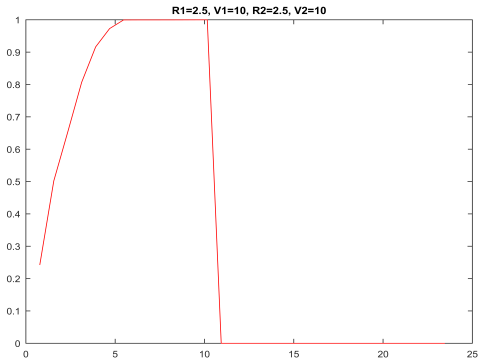

R1=2, V1=10, R2=2.5, V2=10


R1=2.5, V1=10, R2=.5, V2=10


R1=2.5, V1=10, R2=1, V2=10

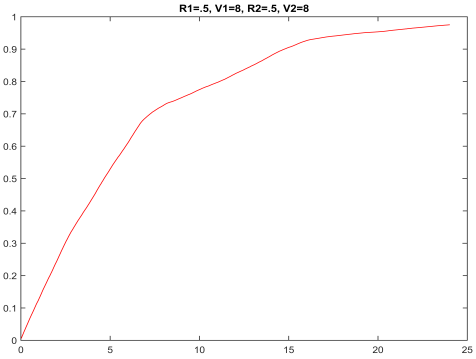

R1=2.5, V1=10, R2=1.5, V2=10

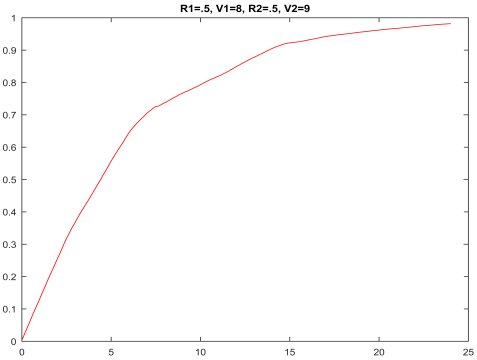

R1=2.5, V1=10, R2=2, V2=10

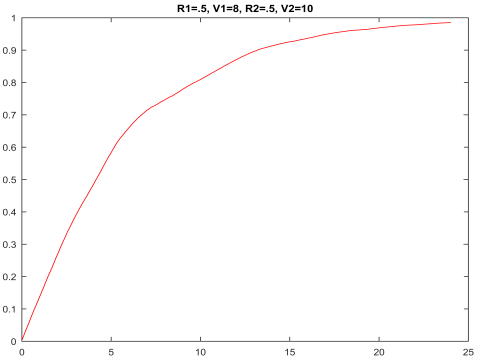

R1=2.5, V1=10, R2=2.5, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

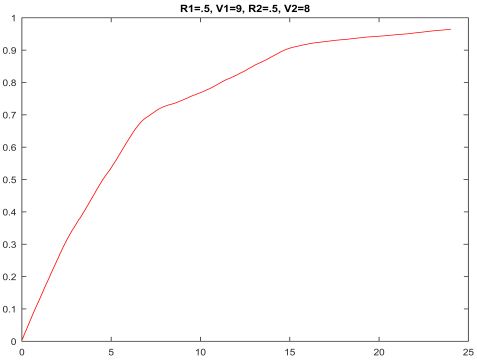
A.1.2 Varying Speed, Constant Search Radii

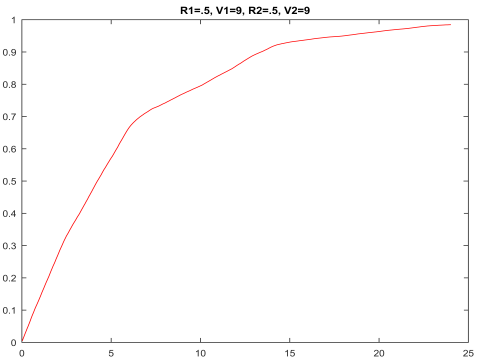
.5nm

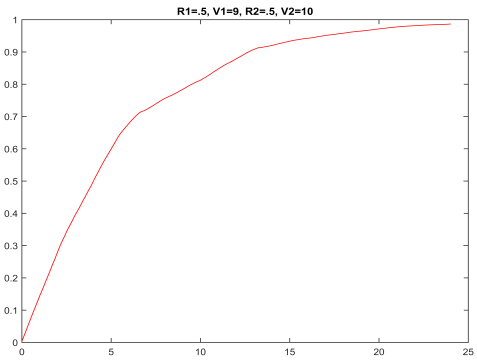


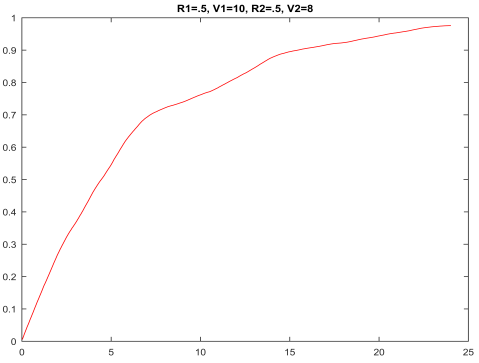


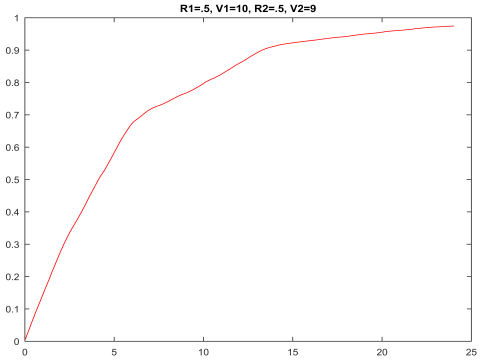


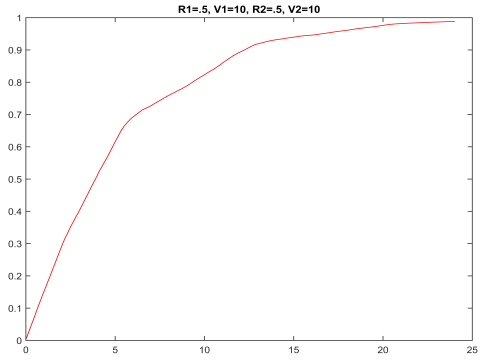




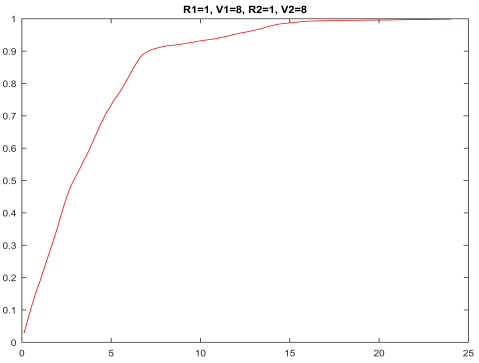


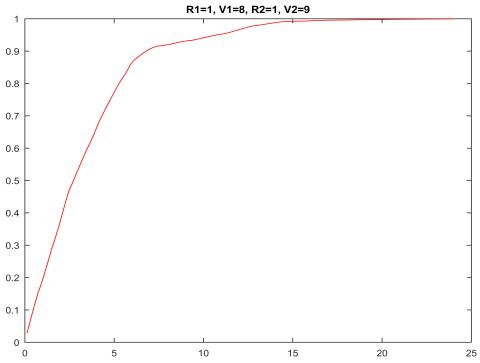

R1=.5, V1=10, R2=.5, V2=8

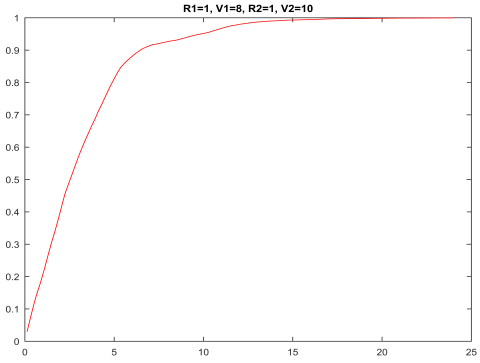

R1=.5, V1=10, R2=.5, V2=9

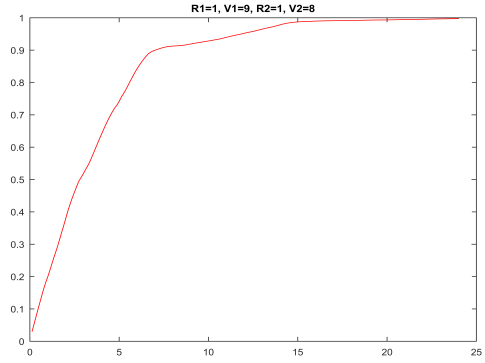

R1=.5, V1=10, R2=.5, V2=10

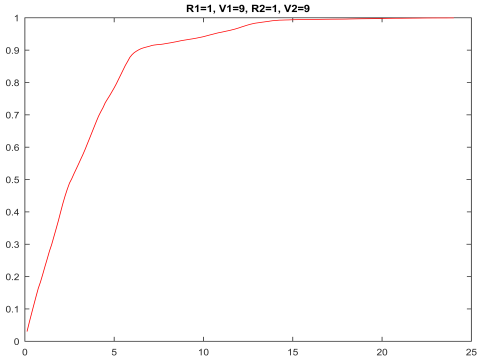
1nm

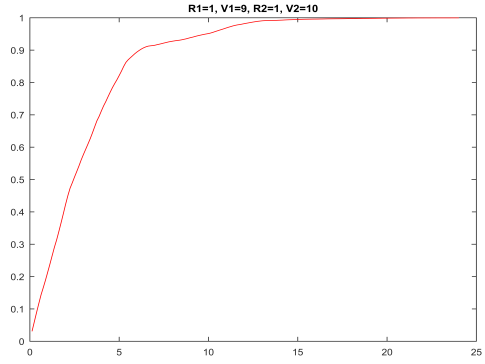

R1=1, V1=8, R2=1, V2=8

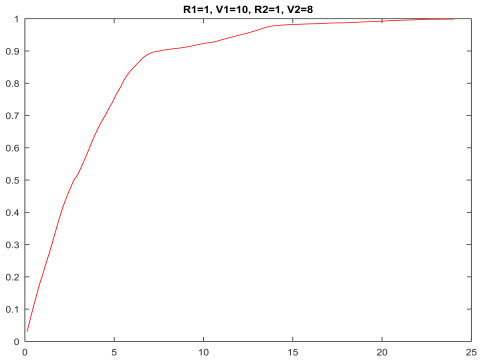

R1=1, V1=8, R2=1, V2=9

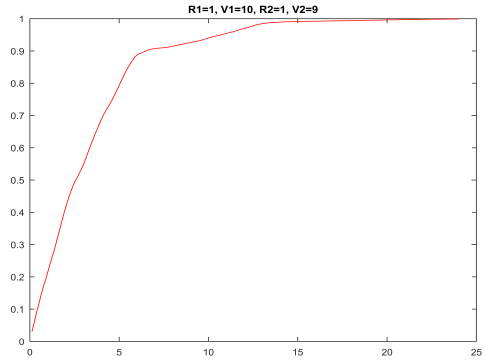

R1=1, V1=8, R2=1, V2=10
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

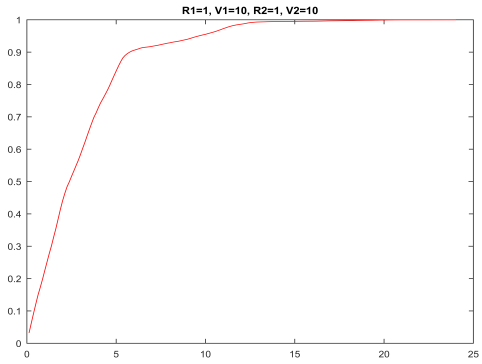

R1=1, V1=9, R2=1, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25


R1=1, V1=9, R2=1, V2=9
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

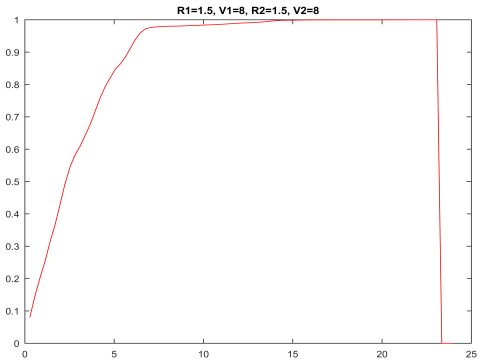

R1=1, V1=9, R2=1, V2=10
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

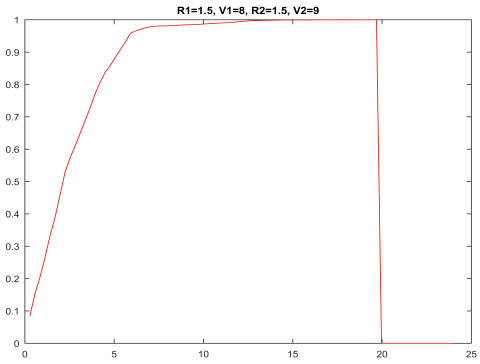

R1=1, V1=10, R2=1, V2=8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

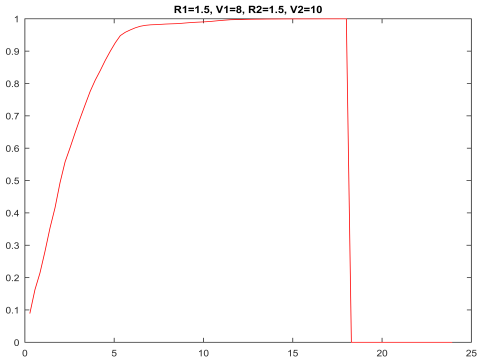

R1=1, V1=10, R2=1, V2=9
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25

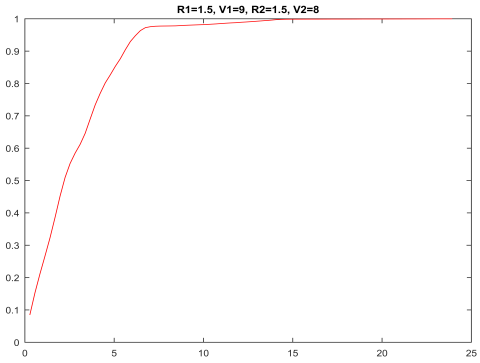

R1=1, V1=10, R2=1, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

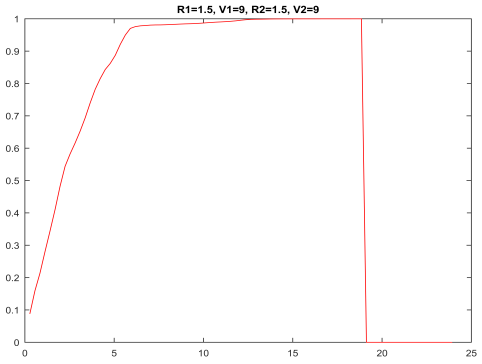
1.5nm

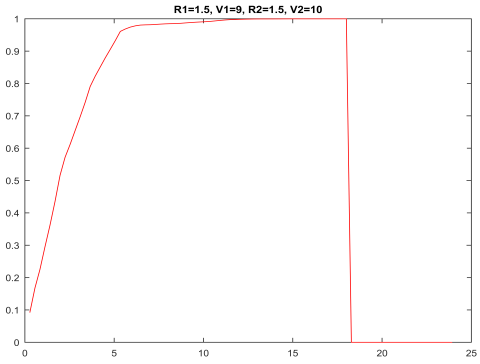

R1=1.5, V1=8, R2=1.5, V2=8

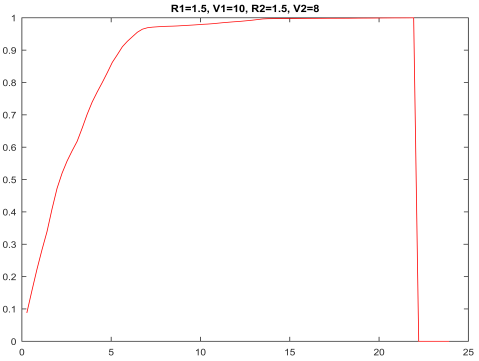

R1=1.5, V1=8, R2=1.5, V2=9

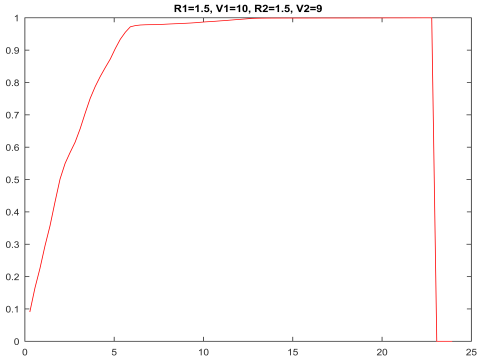

R1=1.5, V1=8, R2=1.5, V2=10

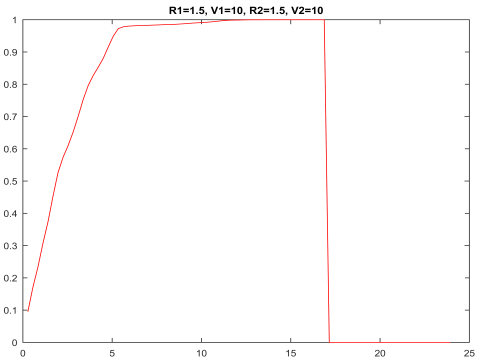

R1=1.5, V1=9, R2=1.5, V2=8


R1=1.5, V1=9, R2=1.5, V2=9

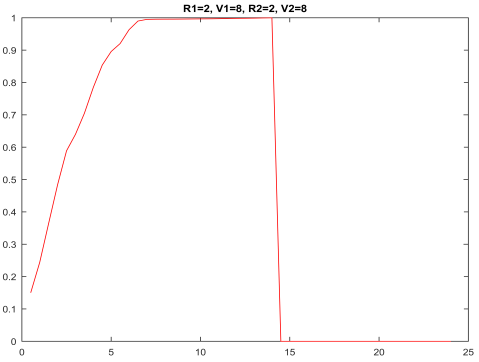

R1=1.5, V1=9, R2=1.5, V2=10

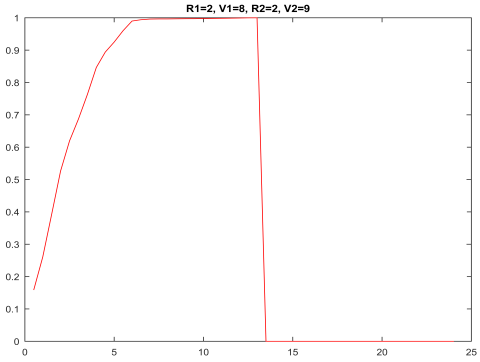

R1=1.5, V1=10, R2=1.5, V2=8

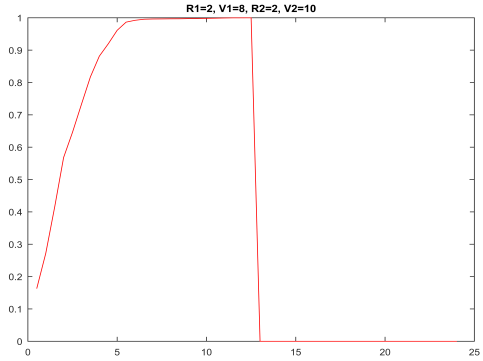

R1=1.5, V1=10, R2=1.5, V2=9

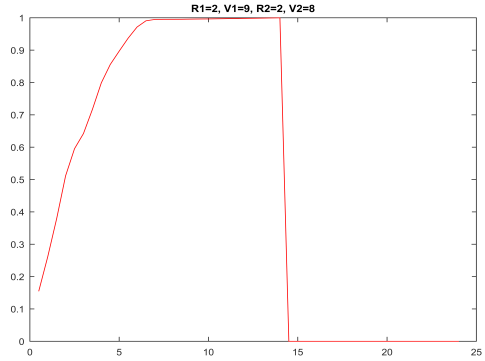

R1=1.5, V1=10, R2=1.5, V2=10

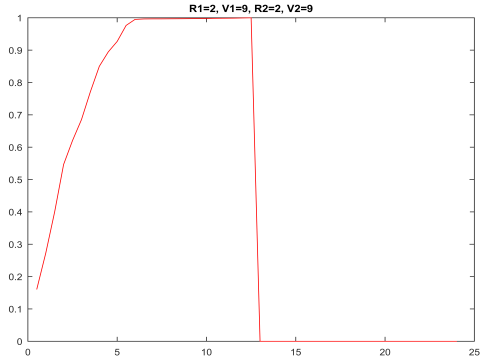
2nm

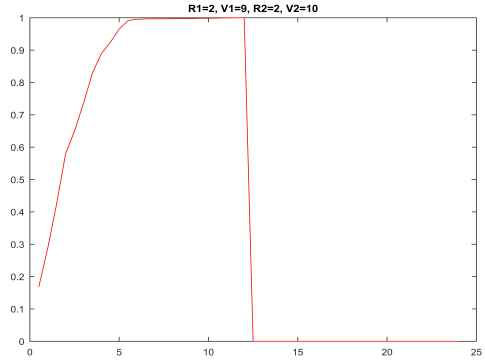

R1=2, V1=8, R2=2, V2=8

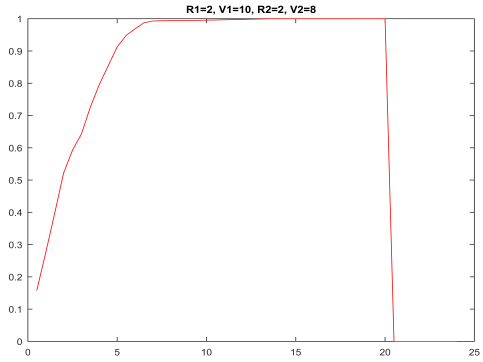

R1=2, V1=8, R2=2, V2=9

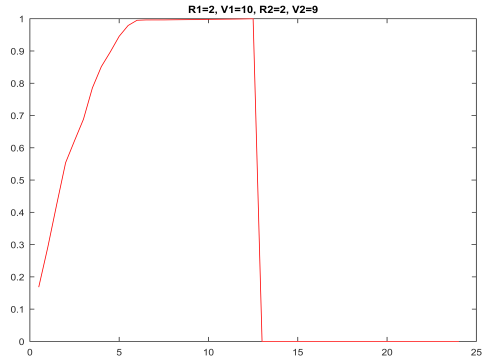

R1=2, V1=8, R2=2, V2=10

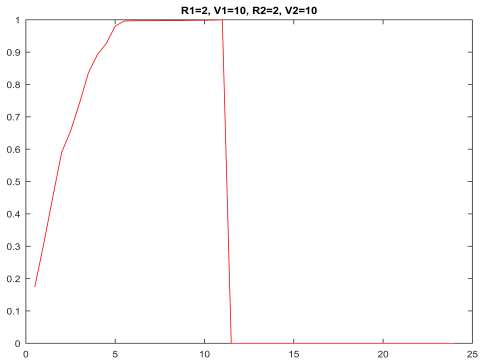

R1=2, V1=9, R2=2, V2=8


R1=2, V1=9, R2=2, V2=9

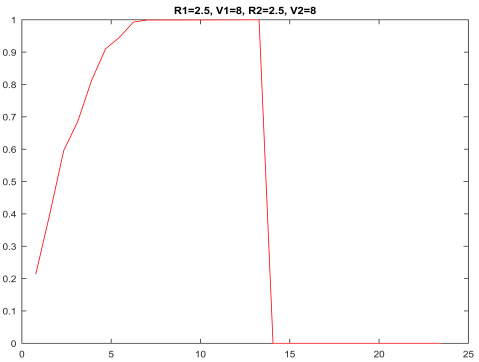

R1=2, V1=9, R2=2, V2=10

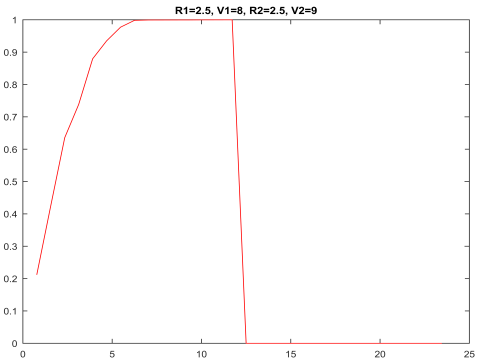

R1=2, V1=10, R2=2, V2=8

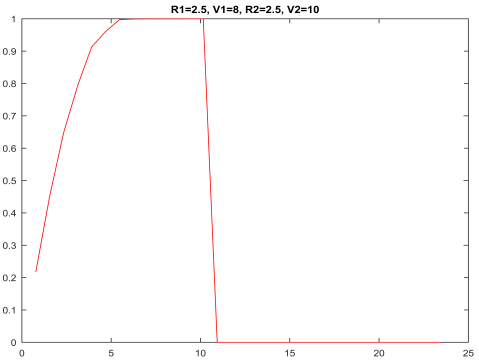

R1=2, V1=10, R2=2, V2=9

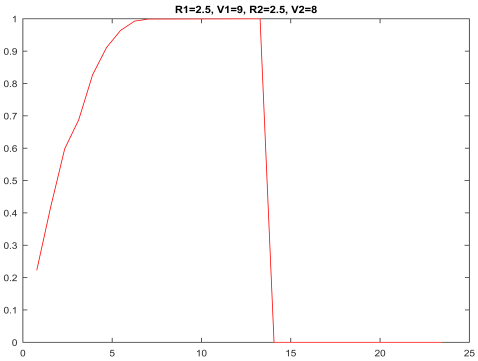

R1=2, V1=10, R2=2, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

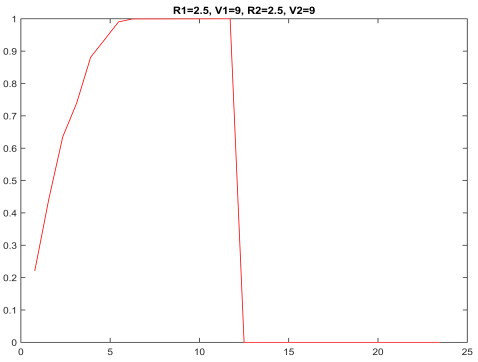
2.5nm

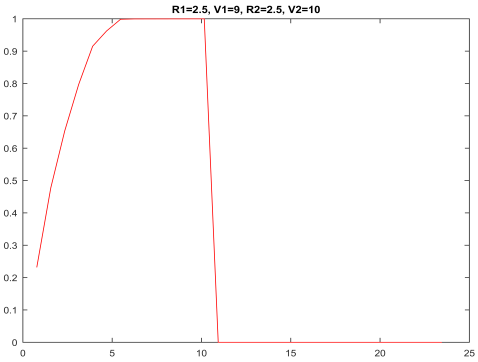

R1=2.5, V1=8, R2=2.5, V2=8

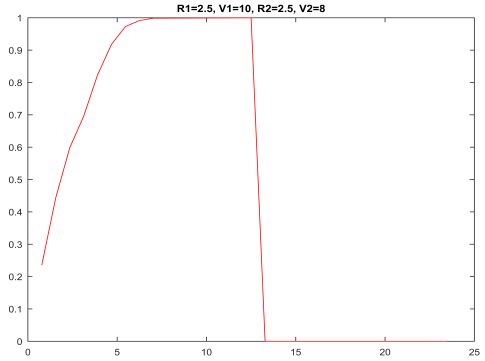

R1=2.5, V1=8, R2=2.5, V2=9

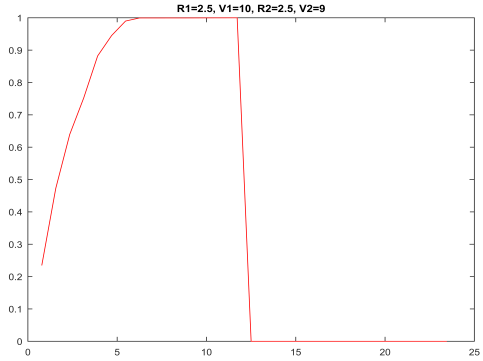

R1=2.5, V1=8, R2=2.5, V2=10

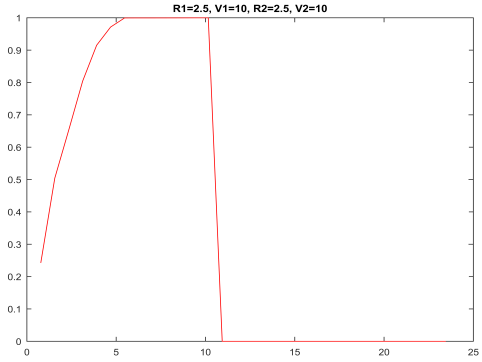

R1=2.5, V1=9, R2=2.5, V2=8


R1=2.5, V1=9, R2=2.5, V2=9


R1=2.5, V1=9, R2=2.5, V2=10





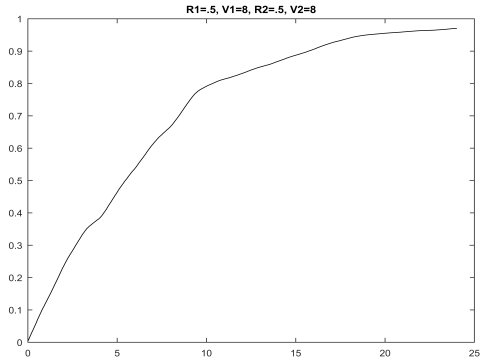


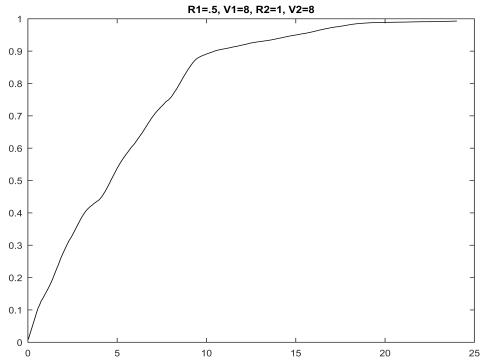
A.2 Multi-Path Ladder Search

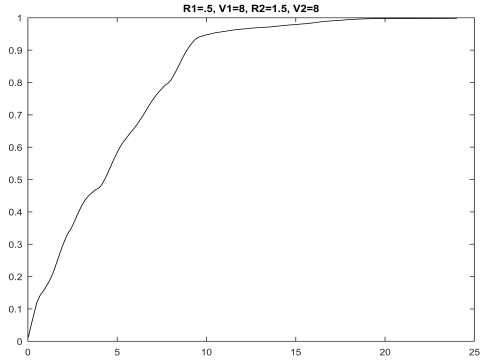
The following figures are for two searchers conducting a multi-path ladder search.

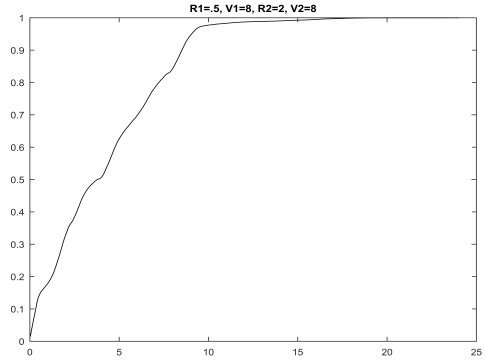
A.2.1 Varying Search Radii, Constant Speed

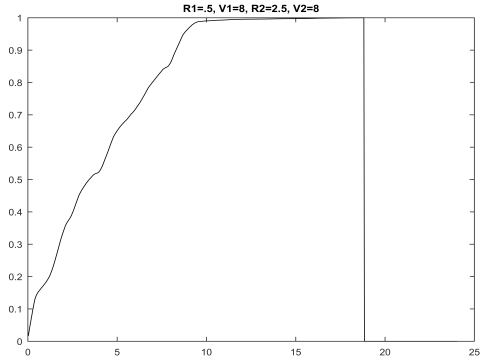
8 kts

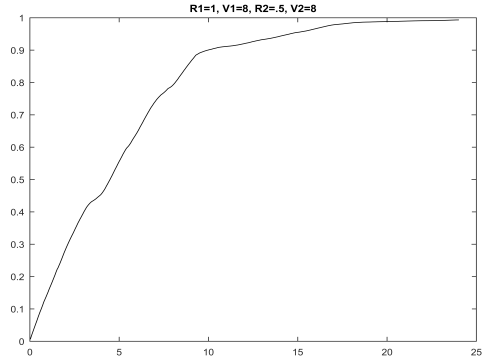


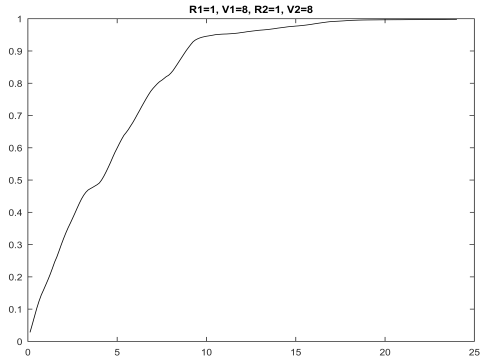


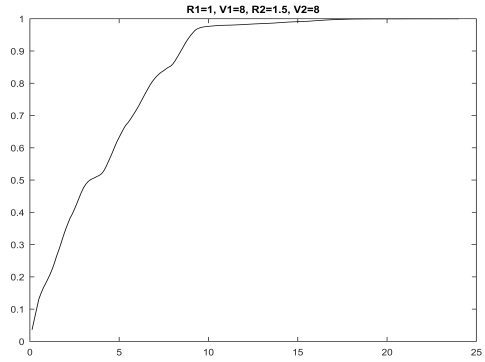

R1=.5, V1=8, R2=1.5, V2=8

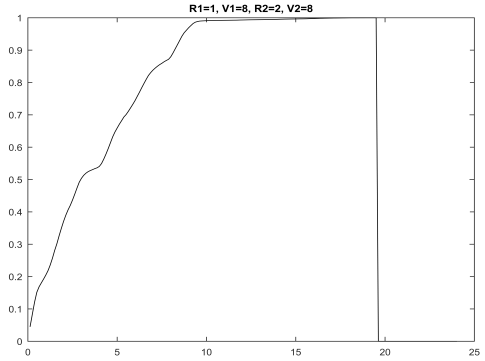

R1=.5, V1=8, R2=2, V2=8

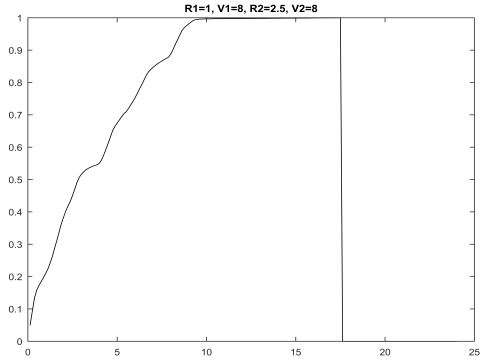

R1=.5, V1=8, R2=2.5, V2=8

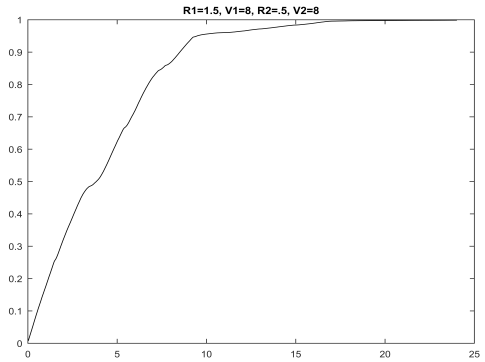

R1=1, V1=8, R2=.5, V2=8

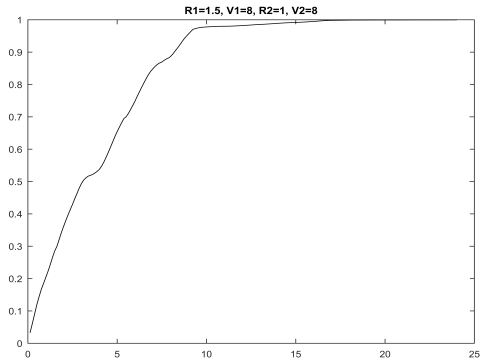

R1=1, V1=8, R2=1, V2=8

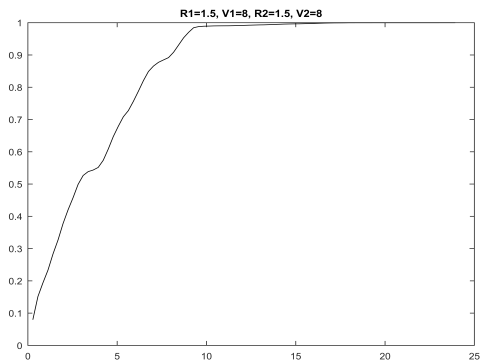

R1=1, V1=8, R2=1.5, V2=8

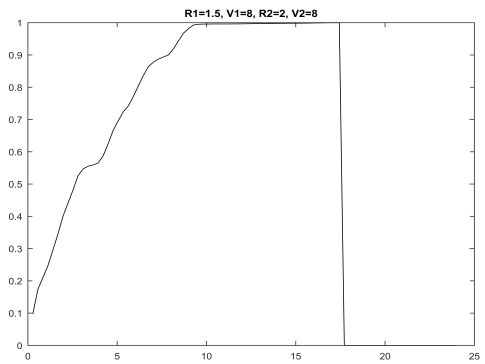

R1=1, V1=8, R2=2, V2=8

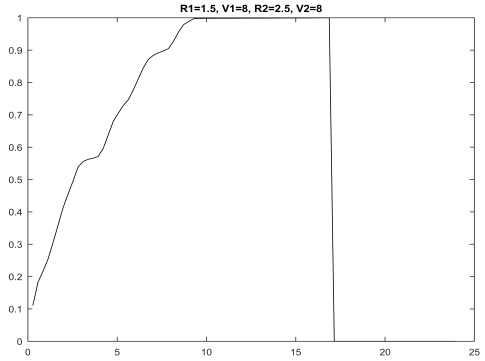

R1=1, V1=8, R2=2.5, V2=8

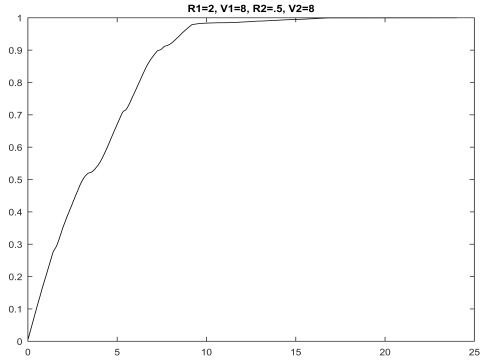

R1=1.5, V1=8, R2=.5, V2=8

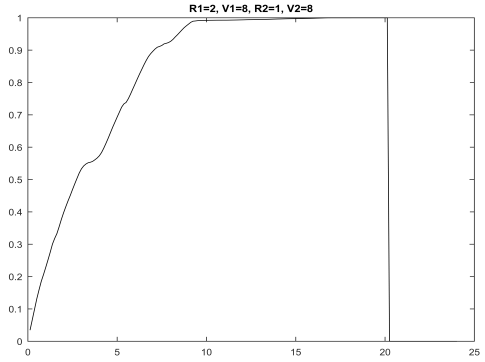

R1=1.5, V1=8, R2=1, V2=8

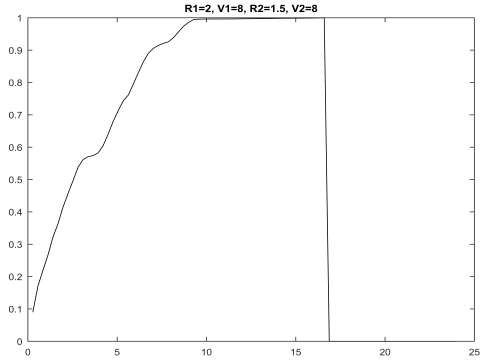

R1=1.5, V1=8, R2=1.5, V2=8

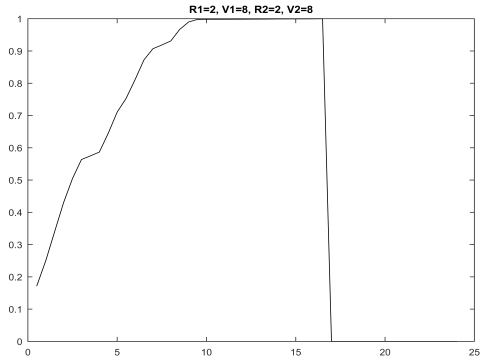

R1=1.5, V1=8, R2=2, V2=8

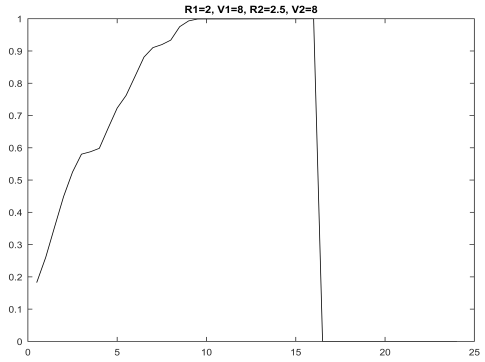

R1=1.5, V1=8, R2=2.5, V2=8

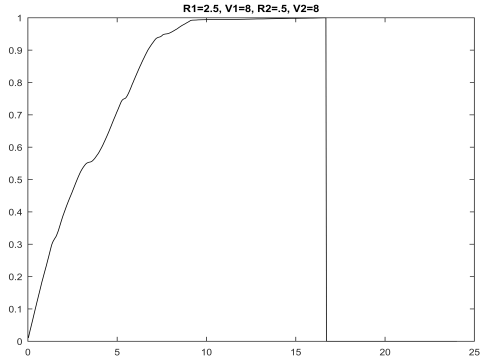

R1=2, V1=8, R2=.5, V2=8

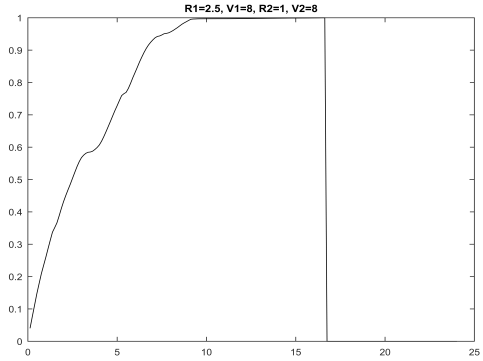

R1=2, V1=8, R2=1, V2=8

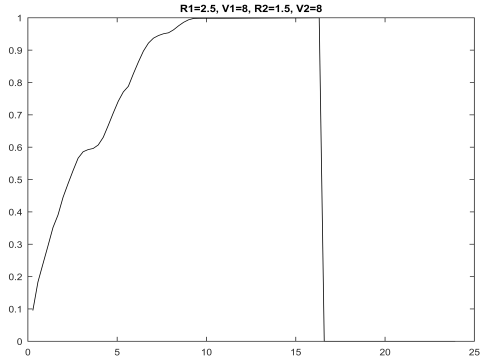

R1=2, V1=8, R2=1.5, V2=8

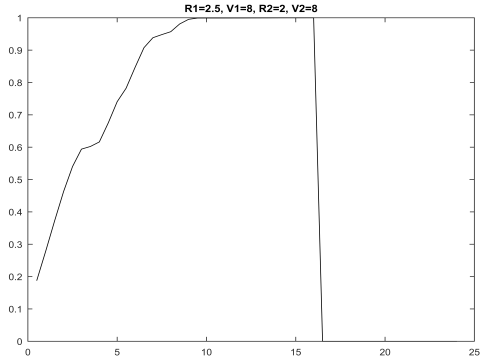

R1=2, V1=8, R2=2, V2=8

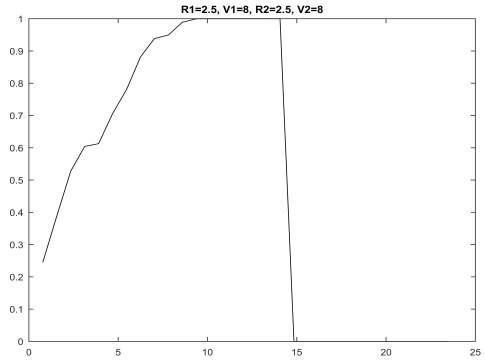

R1=2, V1=8, R2=2.5, V2=8


R1=2.5, V1=8, R2=.5, V2=8

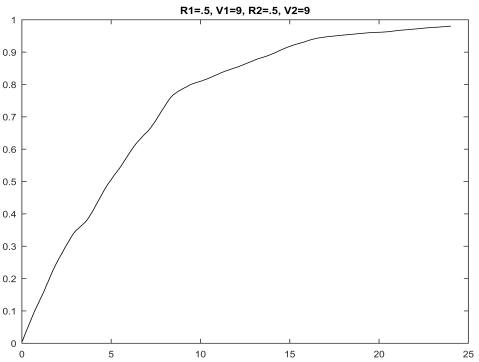

R1=2.5, V1=8, R2=1, V2=8

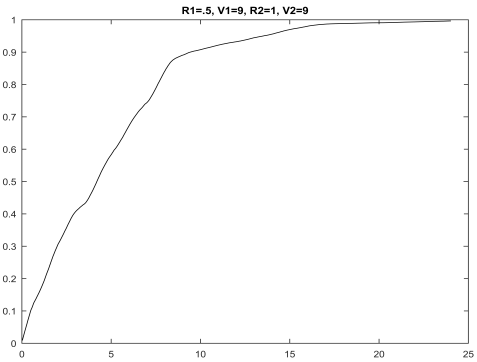

R1=2.5, V1=8, R2=1.5, V2=8

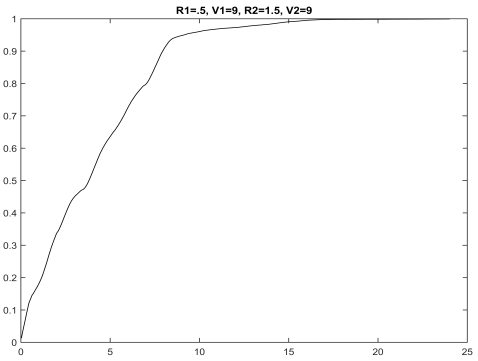

R1=2.5, V1=8, R2=2, V2=8

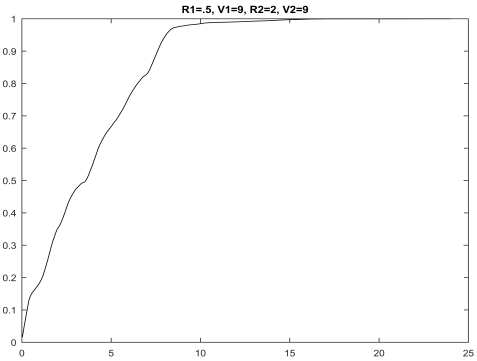

R1=2.5, V1=8, R2=2.5, V2=8

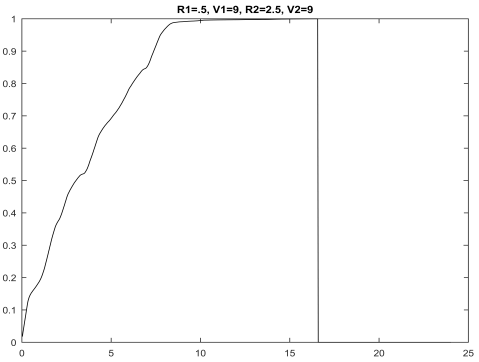
9 kts

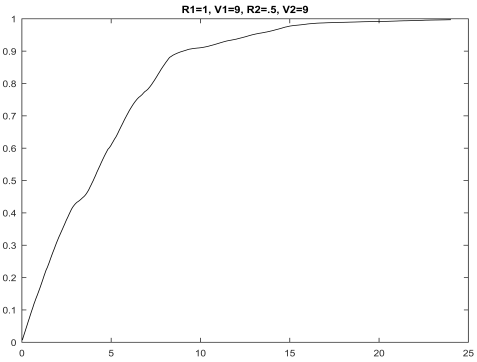

R1=.5, V1=9, R2=.5, V2=9

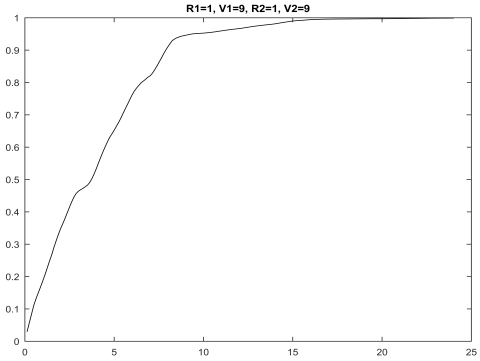

R1=.5, V1=9, R2=1, V2=9

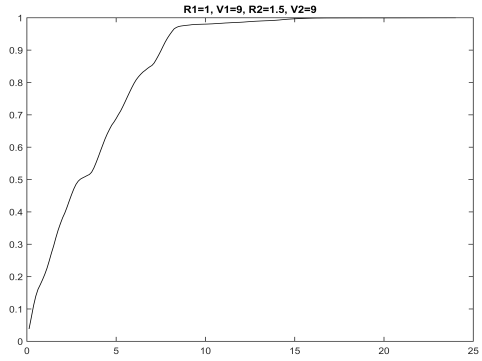

R1=.5, V1=9, R2=1.5, V2=9

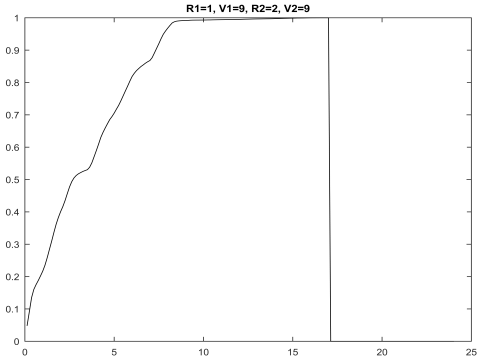

R1=.5, V1=9, R2=2, V2=9

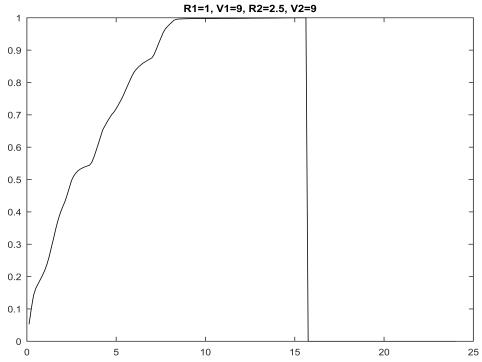

R1=.5, V1=9, R2=2.5, V2=9

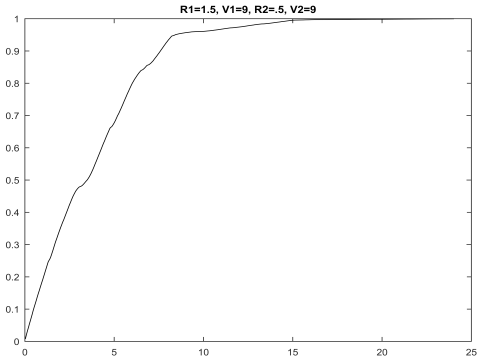

R1=1, V1=9, R2=.5, V2=9

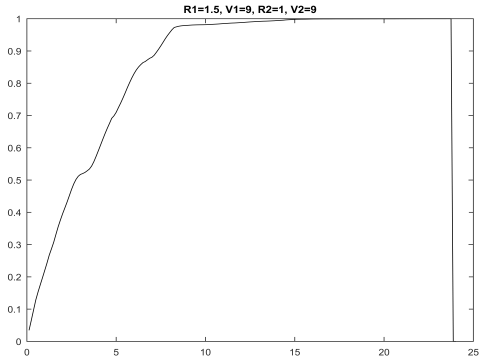

R1=1, V1=9, R2=1, V2=9

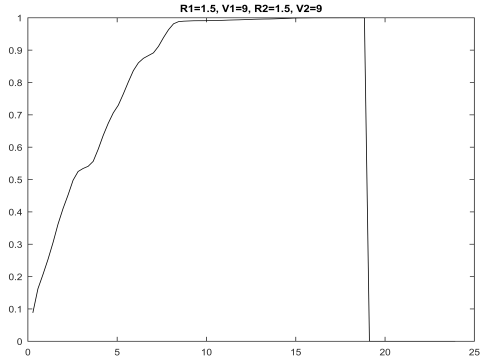

R1=1, V1=9, R2=1.5, V2=9

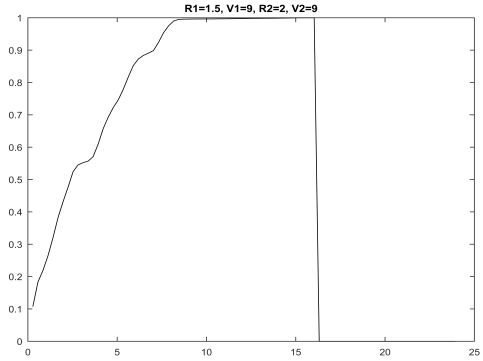

R1=1, V1=9, R2=2, V2=9

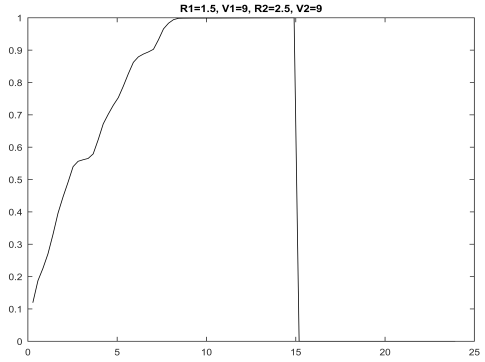

R1=1, V1=9, R2=2.5, V2=9

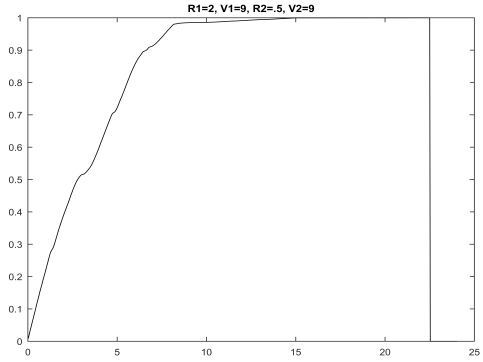

R1=1.5, V1=9, R2=.5, V2=9

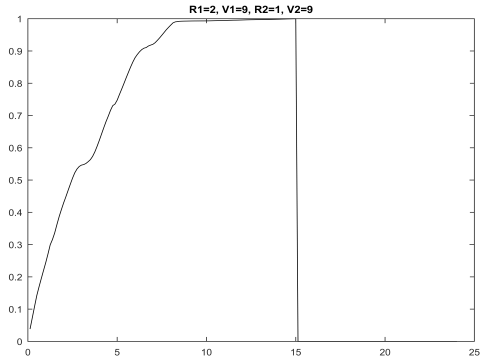

R1=1.5, V1=9, R2=1, V2=9

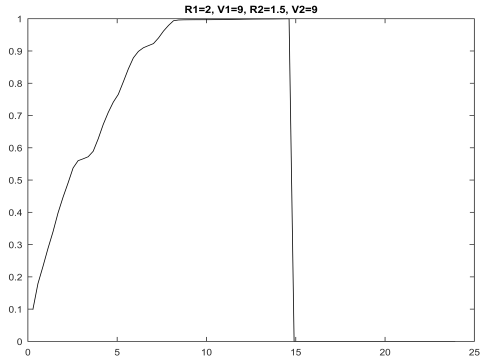

R1=1.5, V1=9, R2=1.5, V2=9

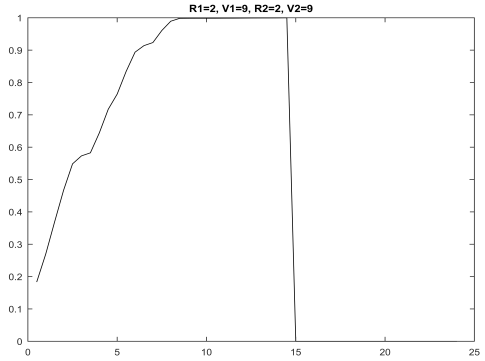

R1=1.5, V1=9, R2=2, V2=9

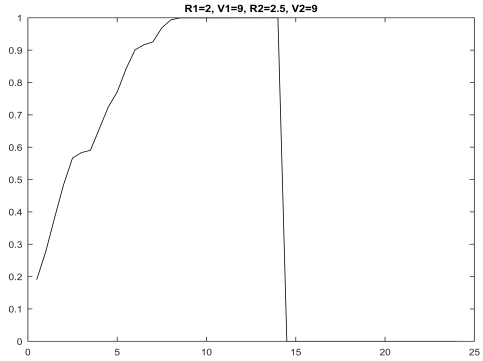

R1=1.5, V1=9, R2=2.5, V2=9

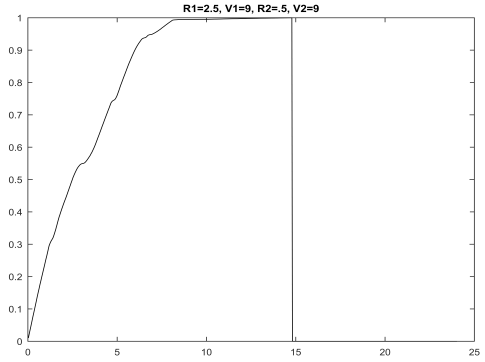

R1=2, V1=9, R2=.5, V2=9

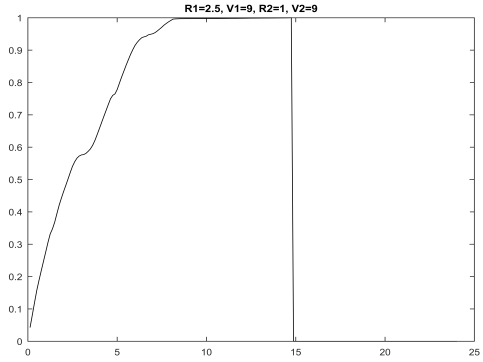

R1=2, V1=9, R2=1, V2=9

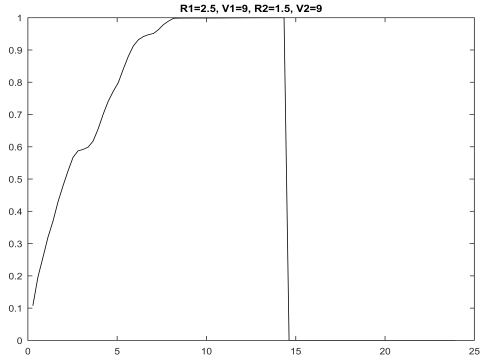

R1=2, V1=9, R2=1.5, V2=9

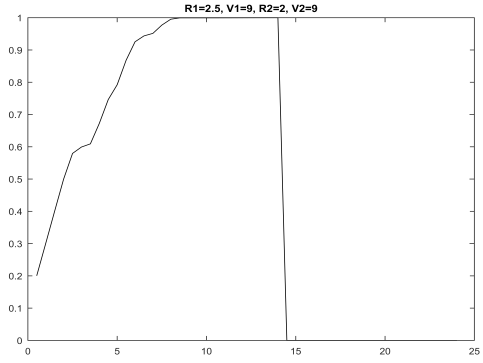

R1=2, V1=9, R2=2, V2=9

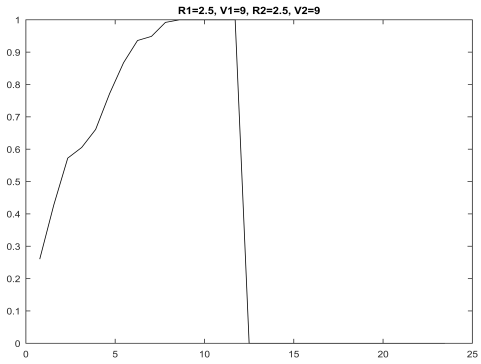

R1=2, V1=9, R2=2.5, V2=9


R1=2.5, V1=9, R2=.5, V2=9

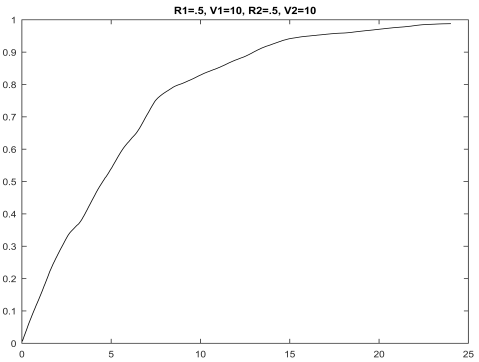

R1=2.5, V1=9, R2=1, V2=9

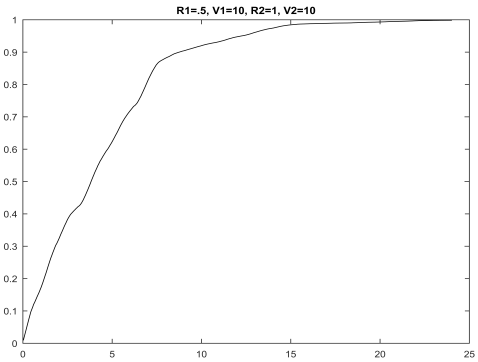

R1=2.5, V1=9, R2=1.5, V2=9

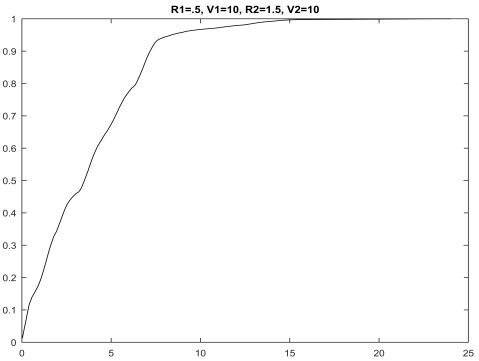

R1=2.5, V1=9, R2=2, V2=9

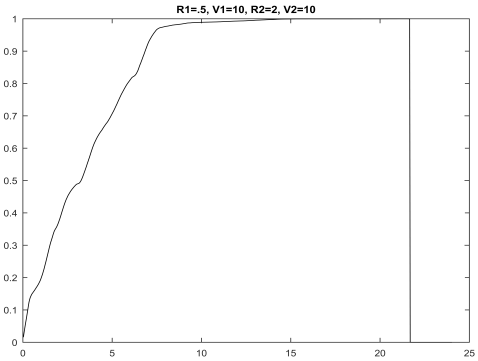

R1=2.5, V1=9, R2=2.5, V2=9
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

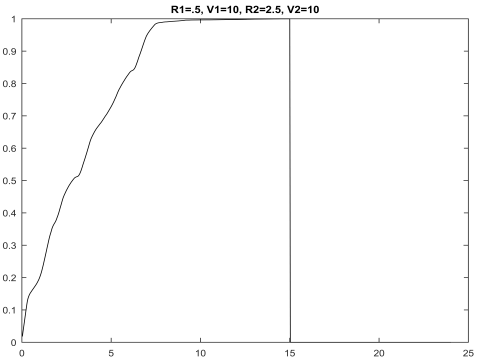
10 kts

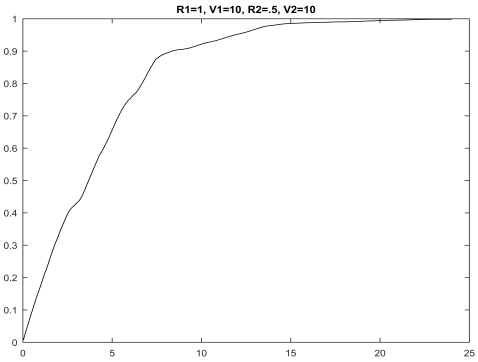

R1=.5, V1=10, R2=.5, V2=10

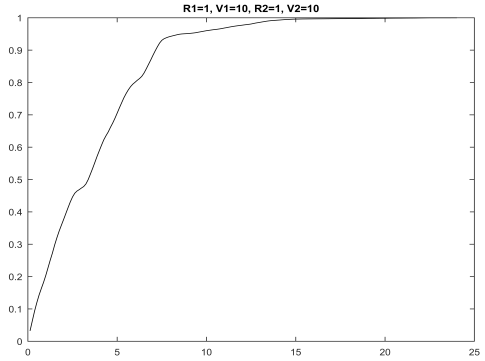

R1=.5, V1=10, R2=1, V2=10

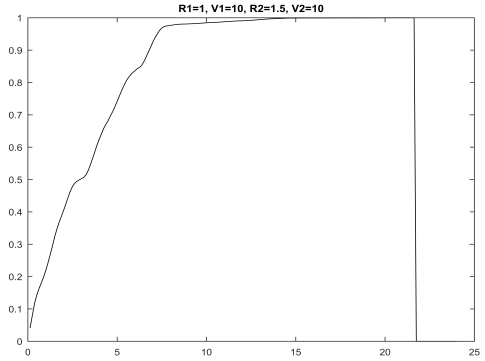

R1=.5, V1=10, R2=1.5, V2=10

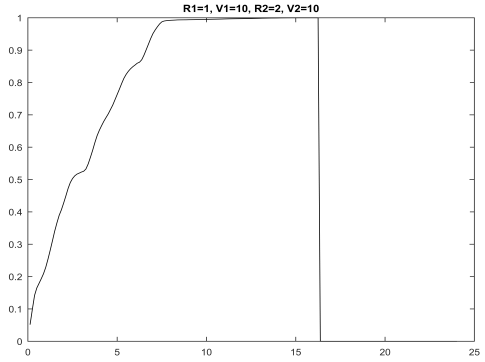

R1=.5, V1=10, R2=2, V2=10

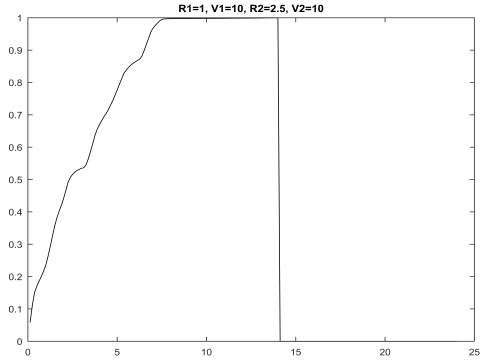

R1=.5, V1=10, R2=2.5, V2=10

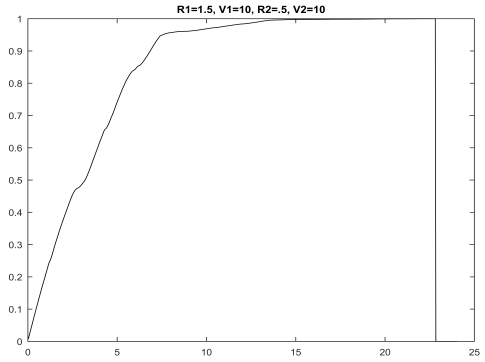

R1=1, V1=10, R2=.5, V2=10

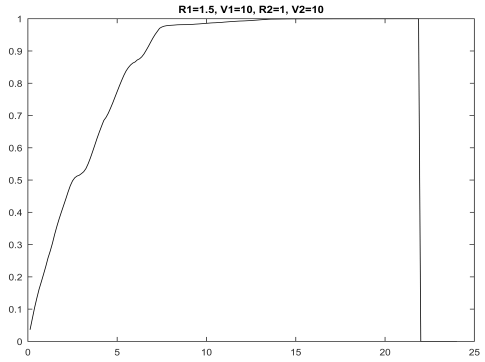

R1=1, V1=10, R2=1, V2=10

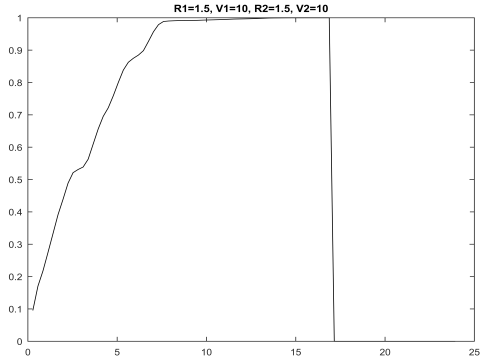

R1=1, V1=10, R2=1.5, V2=10

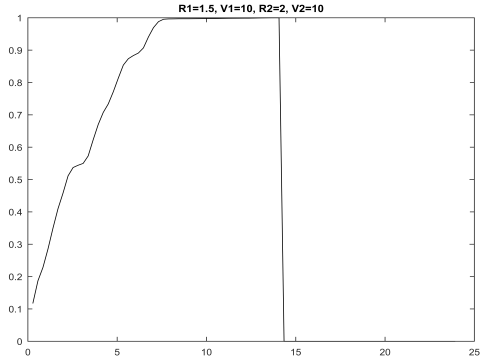

R1=1, V1=10, R2=2, V2=10

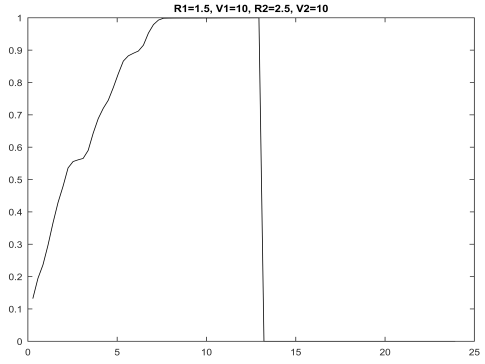

R1=1, V1=10, R2=2.5, V2=10

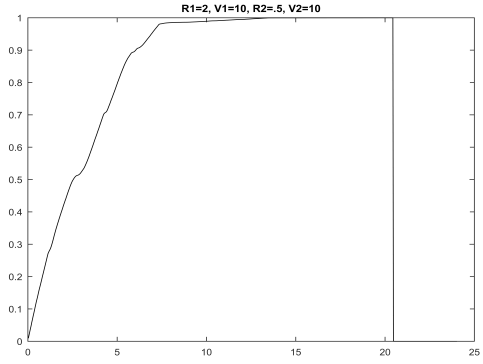

R1=1.5, V1=10, R2=.5, V2=10

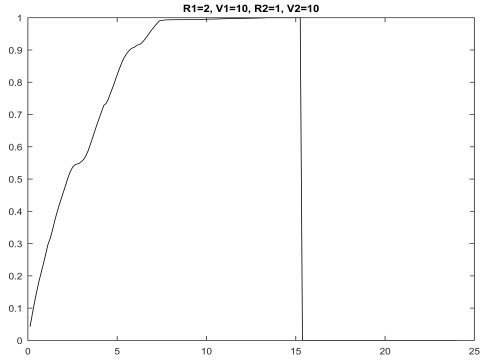

R1=1.5, V1=10, R2=1, V2=10

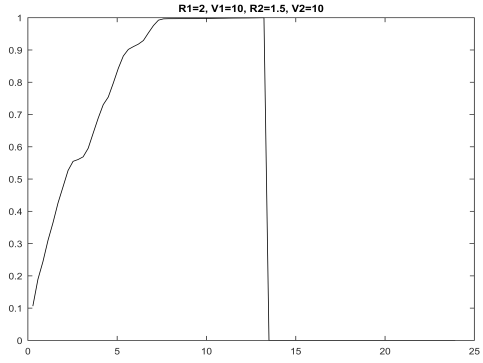

R1=1.5, V1=10, R2=1.5, V2=10

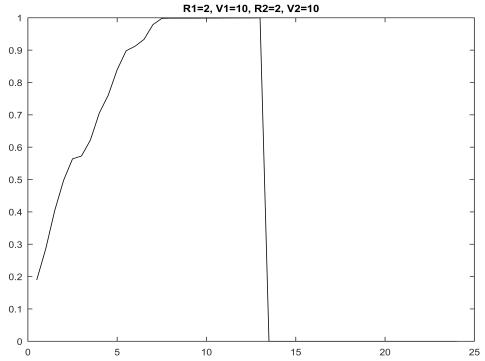

R1=1.5, V1=10, R2=2, V2=10

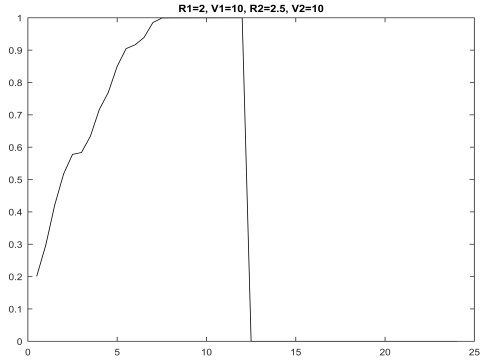

R1=1.5, V1=10, R2=2.5, V2=10

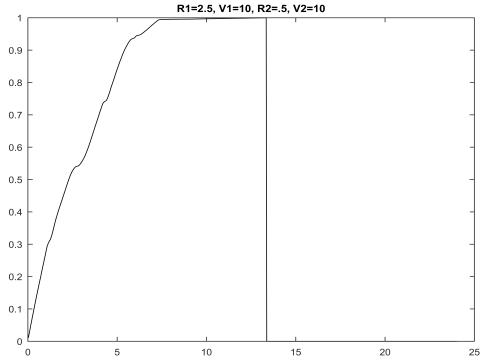

R1=2, V1=10, R2=.5, V2=10

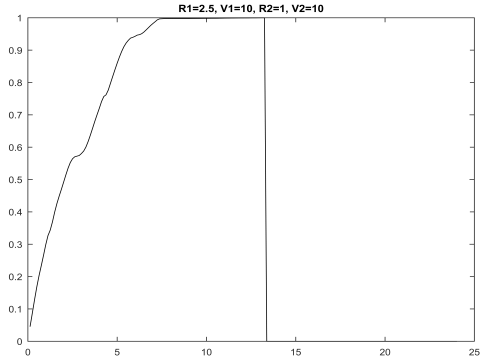

R1=2, V1=10, R2=1, V2=10

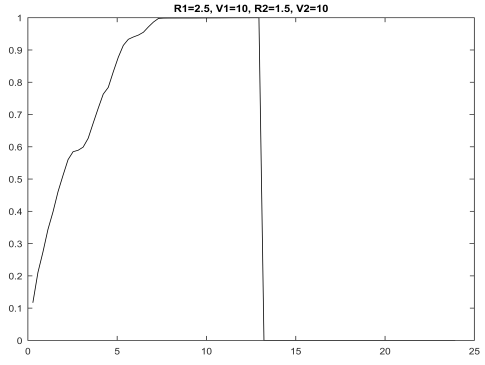

R1=2, V1=10, R2=1.5, V2=10

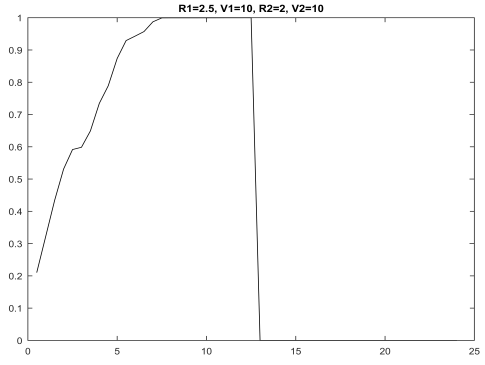

R1=2, V1=10, R2=2, V2=10

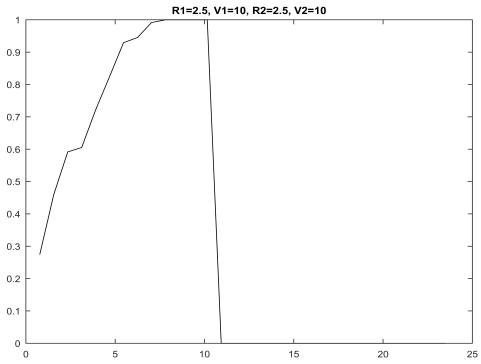

R1=2, V1=10, R2=2.5, V2=10


R1=2.5, V1=10, R2=.5, V2=10


R1=2.5, V1=10, R2=1, V2=10

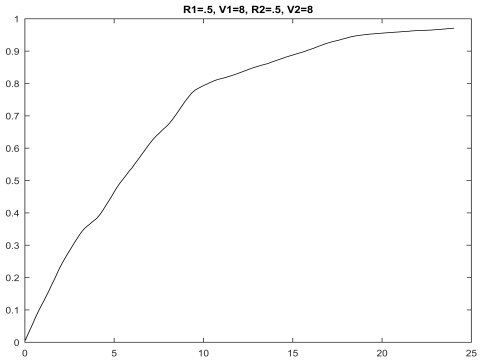

R1=2.5, V1=10, R2=1.5, V2=10

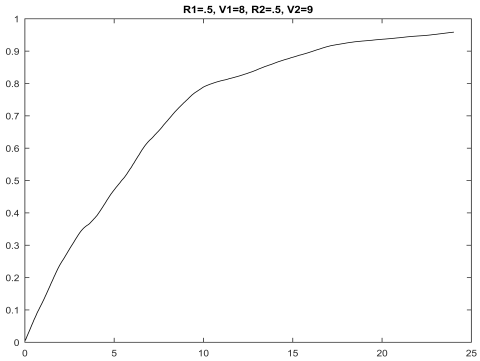

R1=2.5, V1=10, R2=2, V2=10

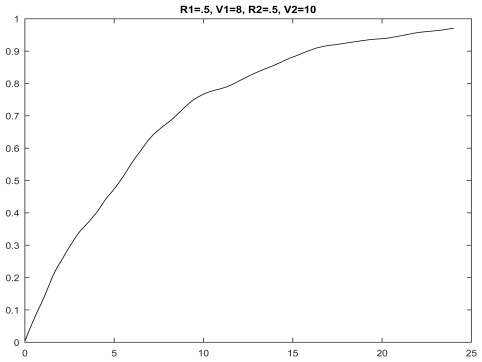

R1=2.5, V1=10, R2=2.5, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

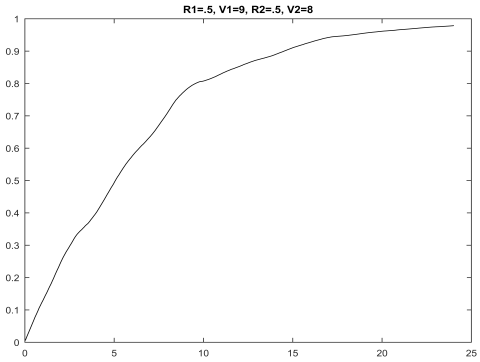
A.2.2 Varying Speed, Constant Search Radii

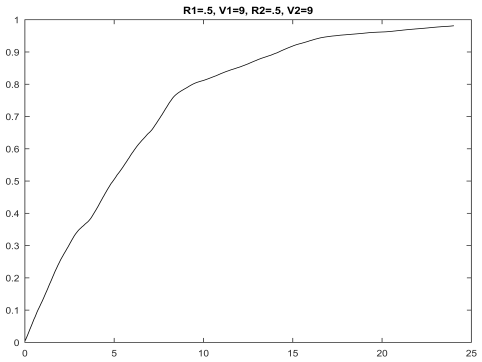
.5nm

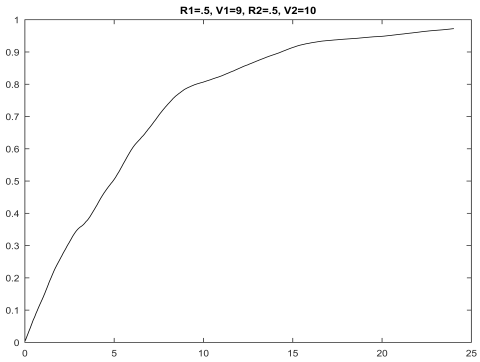


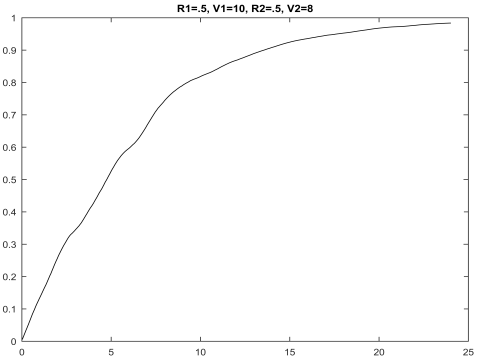


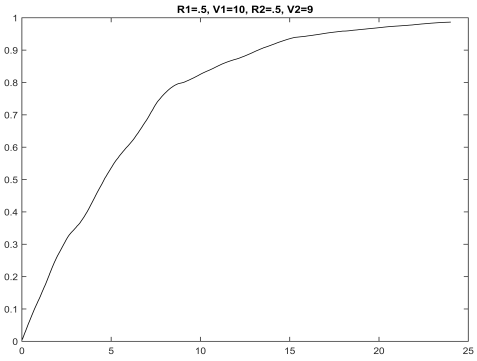


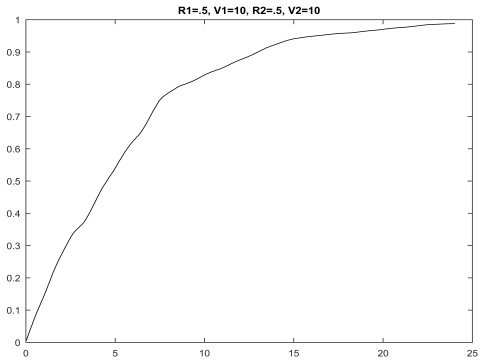




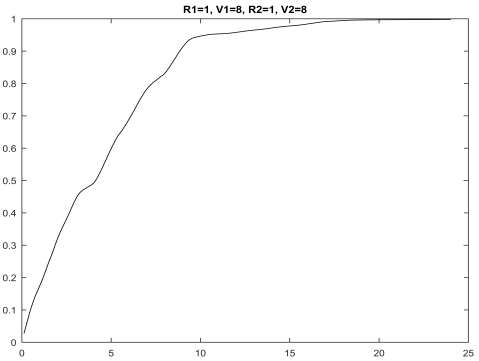


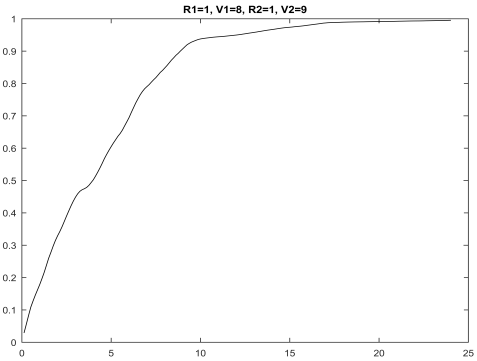

R1=.5, V1=10, R2=.5, V2=8

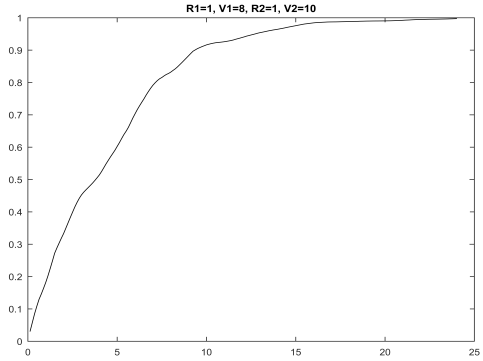

R1=.5, V1=10, R2=.5, V2=9

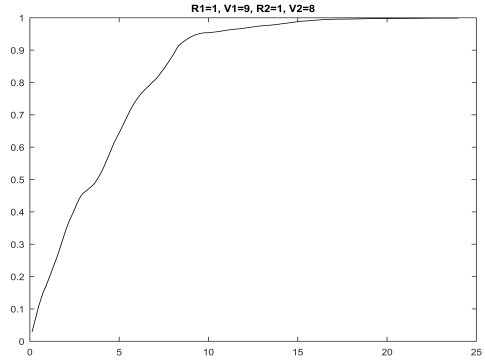

R1=.5, V1=10, R2=.5, V2=10

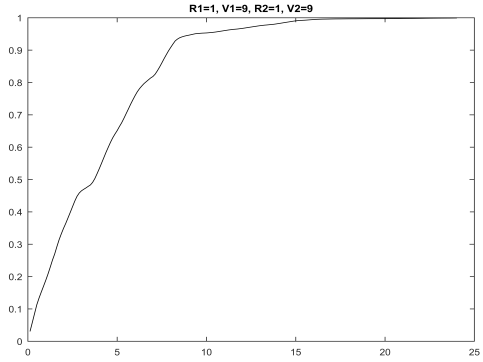
1nm

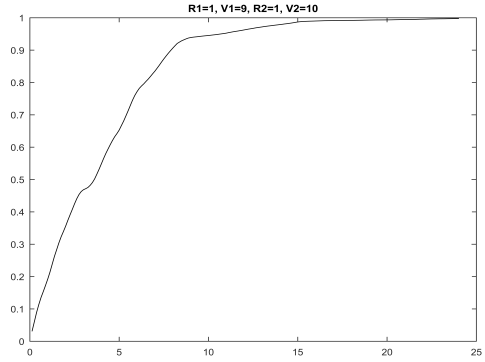

R1=1, V1=8, R2=1, V2=8

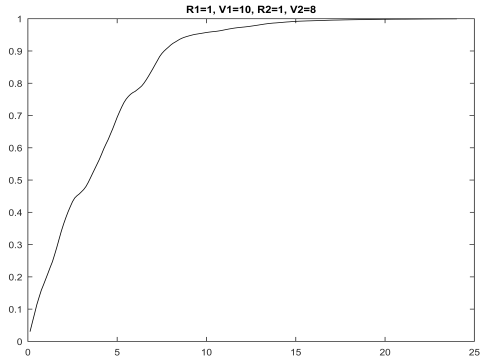

R1=1, V1=8, R2=1, V2=9

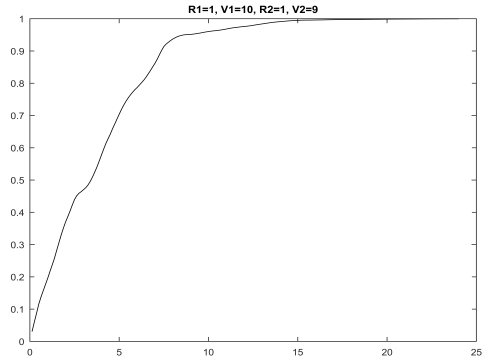

R1=1, V1=8, R2=1, V2=10

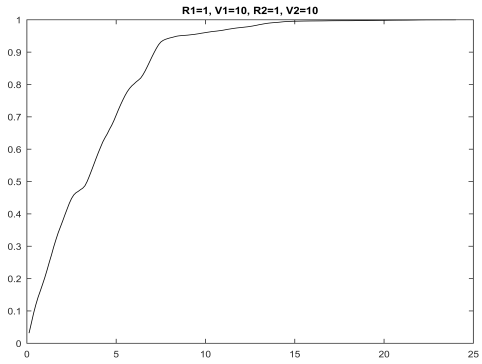

R1=1, V1=9, R2=1, V2=8


R1=1, V1=9, R2=1, V2=9

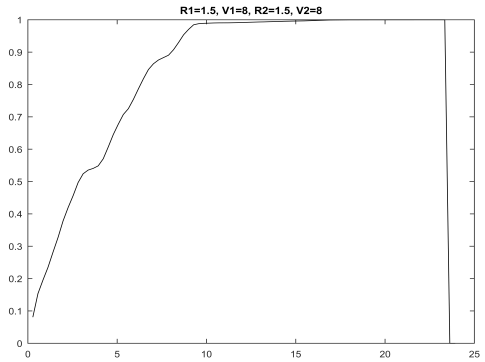

R1=1, V1=9, R2=1, V2=10

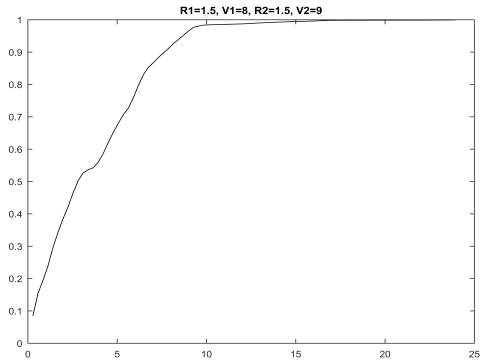

R1=1, V1=10, R2=1, V2=8

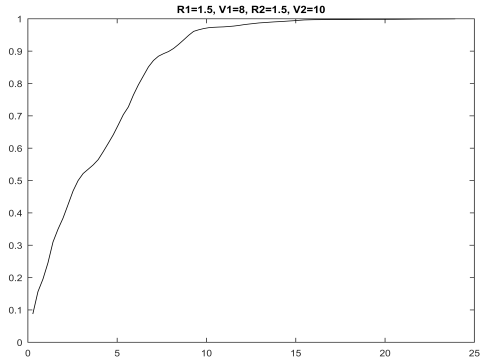

R1=1, V1=10, R2=1, V2=9

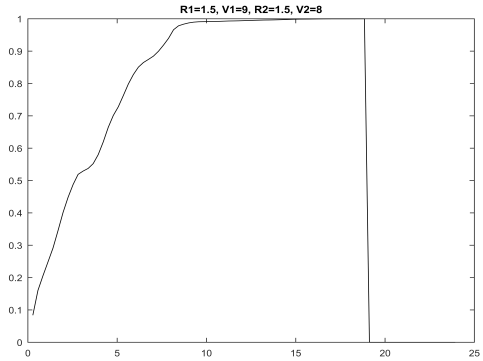

R1=1, V1=10, R2=1, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

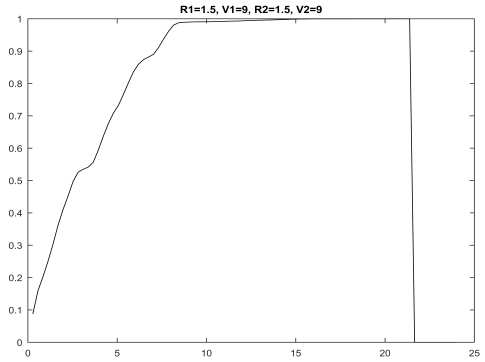
1.5nm

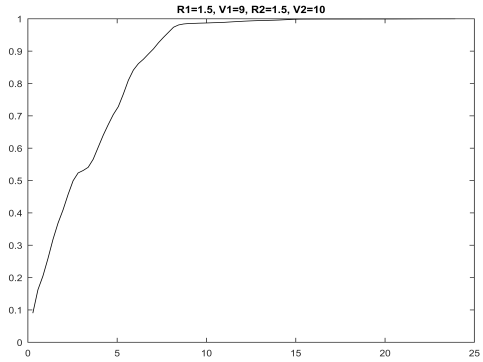

R1=1.5, V1=8, R2=1.5, V2=8

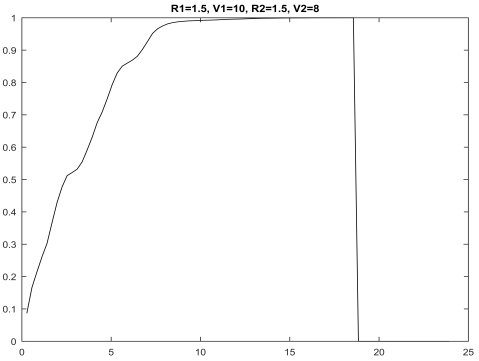

R1=1.5, V1=8, R2=1.5, V2=9

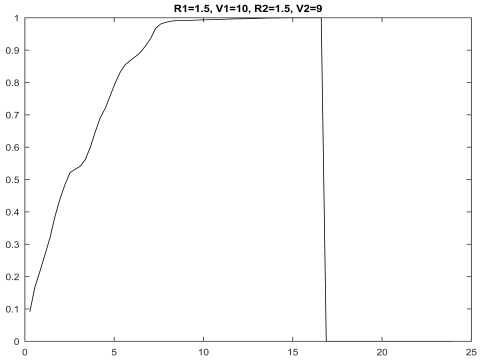

R1=1.5, V1=8, R2=1.5, V2=10

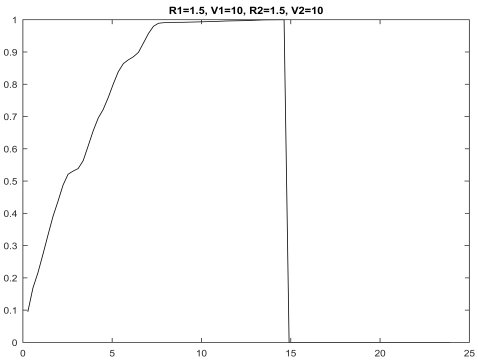

R1=1.5, V1=9, R2=1.5, V2=8


R1=1.5, V1=9, R2=1.5, V2=9

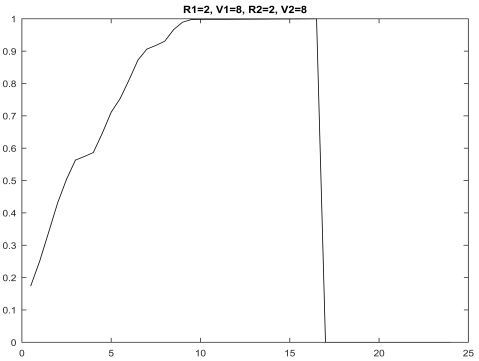

R1=1.5, V1=9, R2=1.5, V2=10

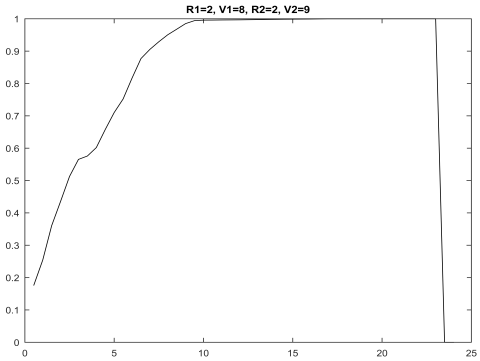

R1=1.5, V1=10, R2=1.5, V2=8

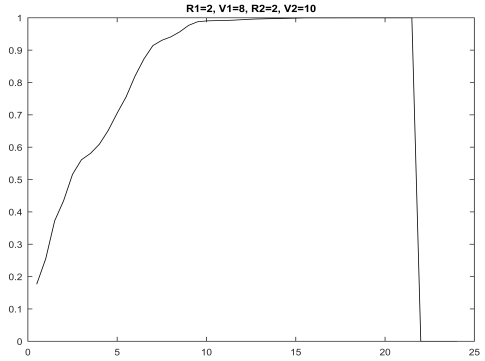

R1=1.5, V1=10, R2=1.5, V2=9

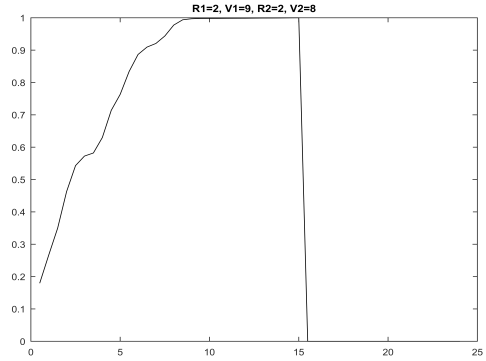

R1=1.5, V1=10, R2=1.5, V2=10

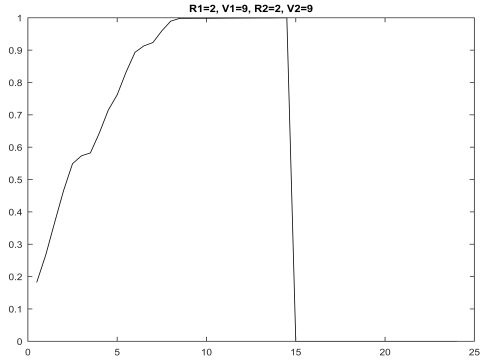
2nm

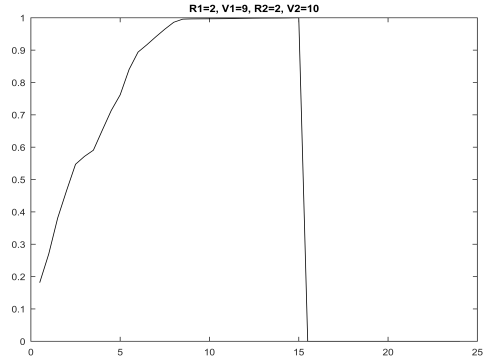

R1=2, V1=8, R2=2, V2=8

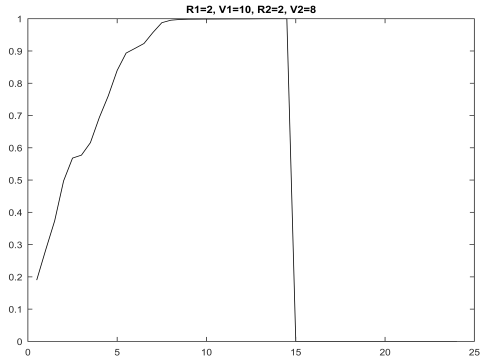

R1=2, V1=8, R2=2, V2=9

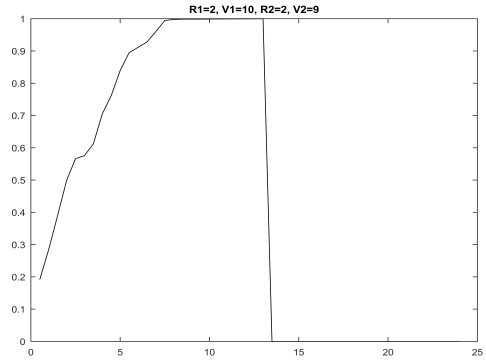

R1=2, V1=8, R2=2, V2=10

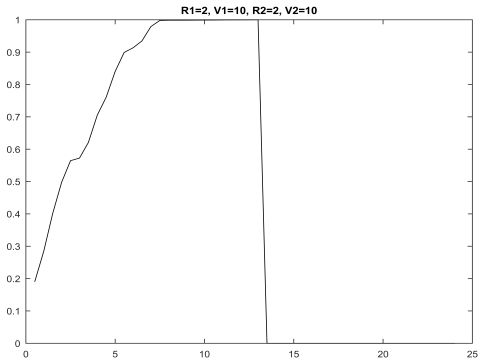

R1=2, V1=9, R2=2, V2=8


R1=2, V1=9, R2=2, V2=9

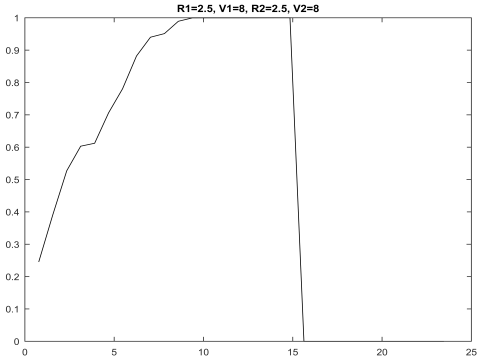

R1=2, V1=9, R2=2, V2=10

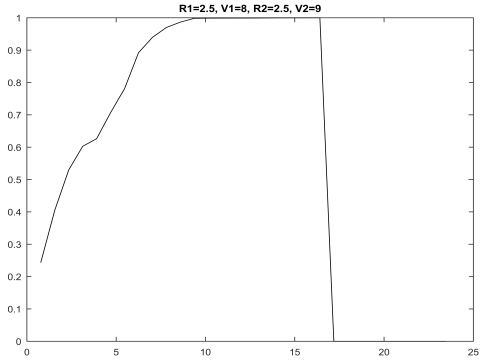

R1=2, V1=10, R2=2, V2=8

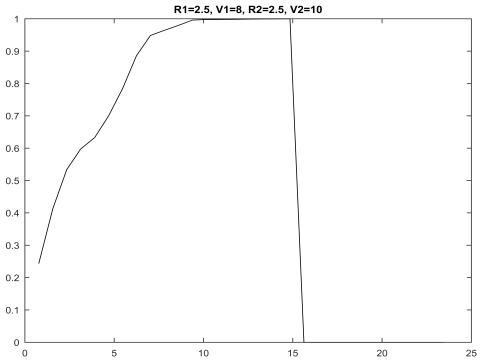

R1=2, V1=10, R2=2, V2=9

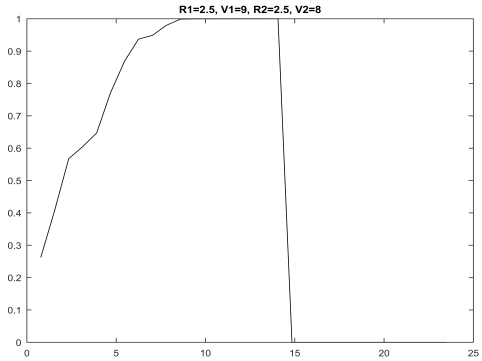

R1=2, V1=10, R2=2, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

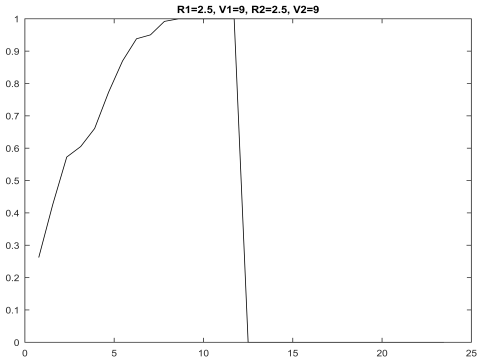
2.5nm

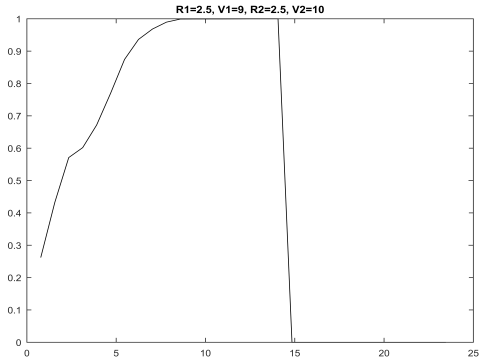

R1=2.5, V1=8, R2=2.5, V2=8

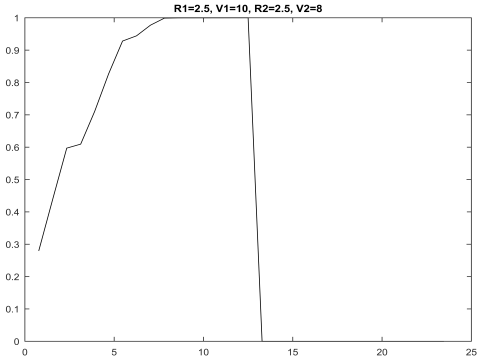

R1=2.5, V1=8, R2=2.5, V2=9

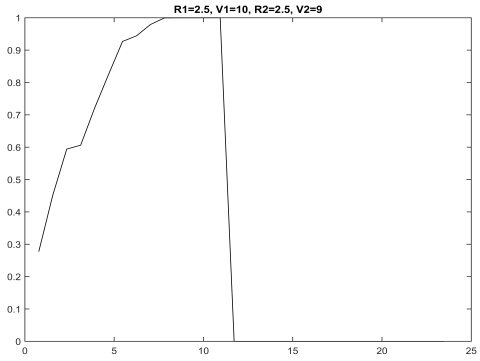

R1=2.5, V1=8, R2=2.5, V2=10

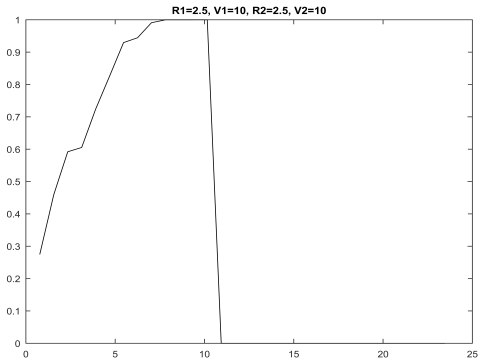

R1=2.5, V1=9, R2=2.5, V2=8


R1=2.5, V1=9, R2=2.5, V2=9


R1=2.5, V1=9, R2=2.5, V2=10





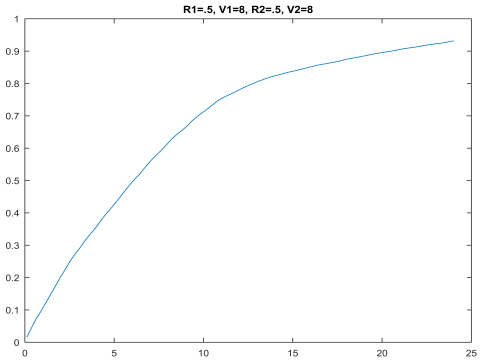


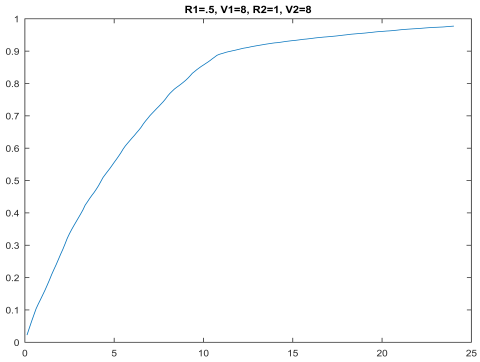
A.3 In-Line Spiral Search

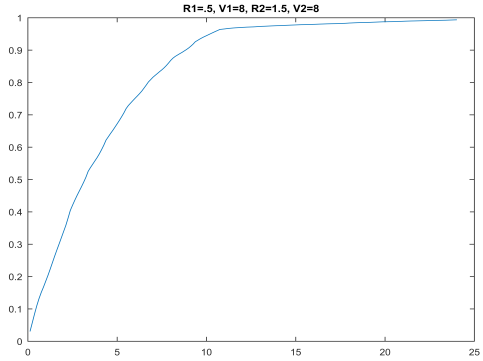
The following figures are for two searchers conducting an in-line spiral search.

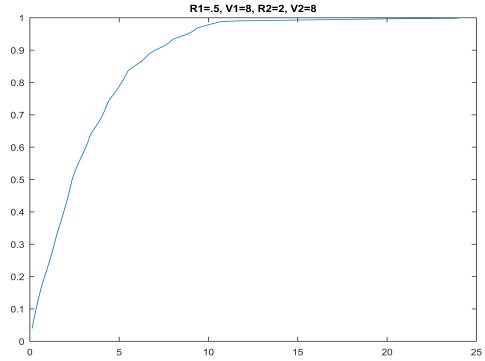
A.3.1 Varying Search Radii, Constant Speed

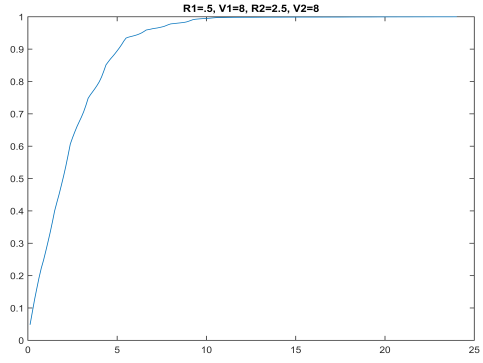
8 kts

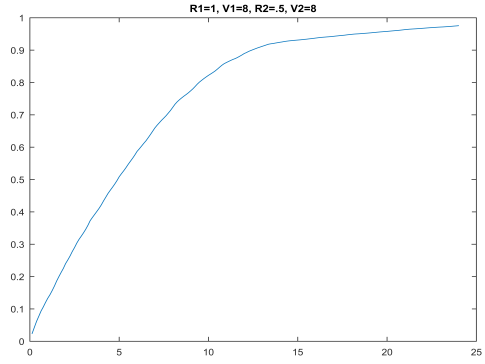


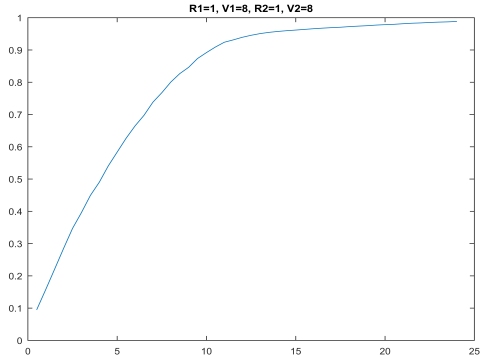


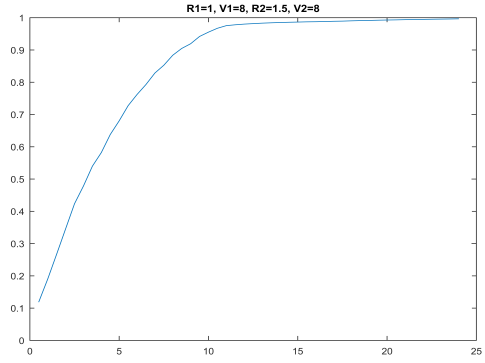

R1=.5, V1=8, R2=1.5, V2=8

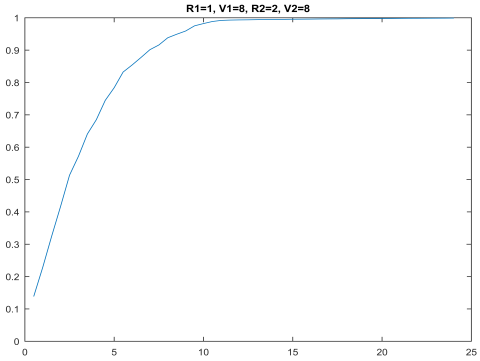

R1=.5, V1=8, R2=2, V2=8

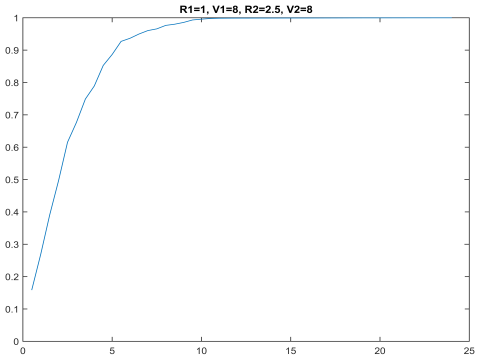

R1=.5, V1=8, R2=2.5, V2=8

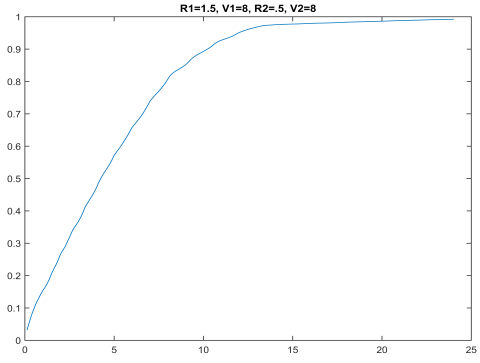

R1=1, V1=8, R2=.5, V2=8

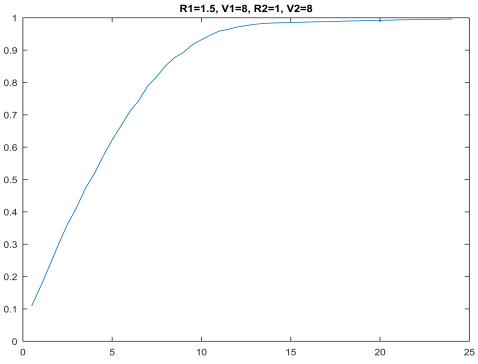

R1=1, V1=8, R2=1, V2=8

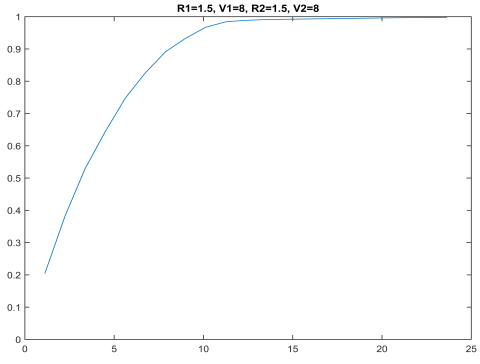

R1=1, V1=8, R2=1.5, V2=8

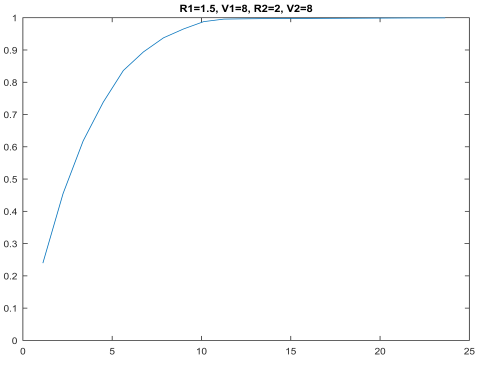

R1=1, V1=8, R2=2, V2=8

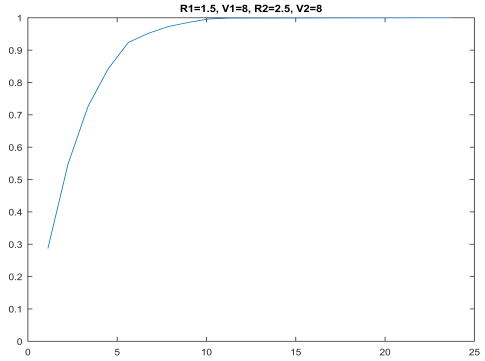

R1=1, V1=8, R2=2.5, V2=8

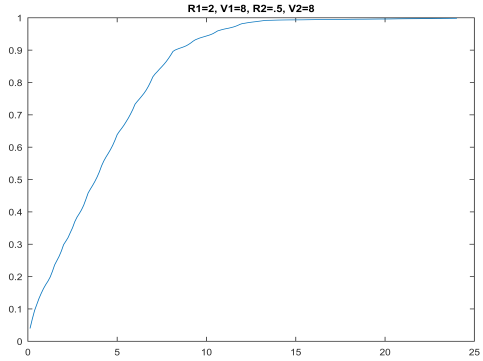

R1=1.5, V1=8, R2=.5, V2=8

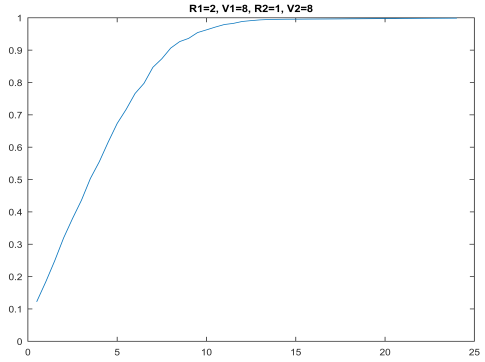

R1=1.5, V1=8, R2=1, V2=8

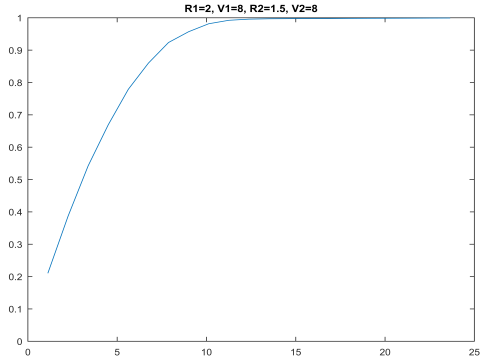

R1=1.5, V1=8, R2=1.5, V2=8

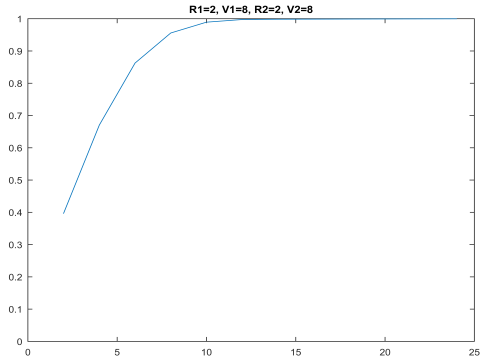

R1=1.5, V1=8, R2=2, V2=8

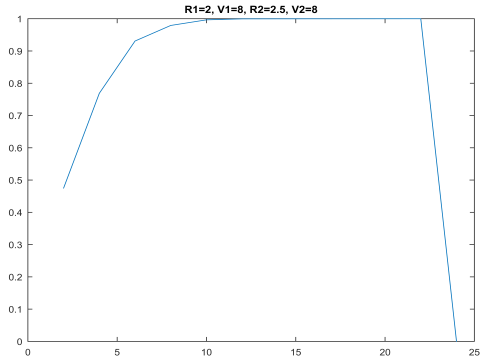

R1=1.5, V1=8, R2=2.5, V2=8

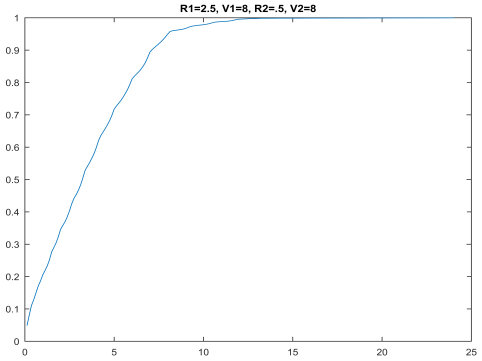

R1=2, V1=8, R2=.5, V2=8

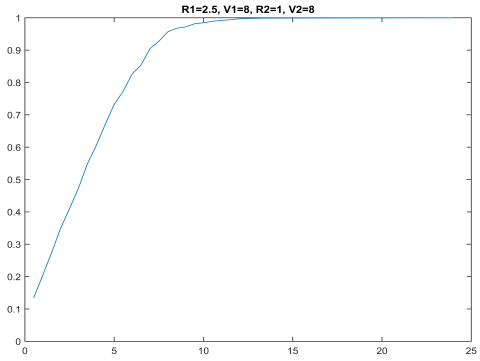

R1=2, V1=8, R2=1, V2=8

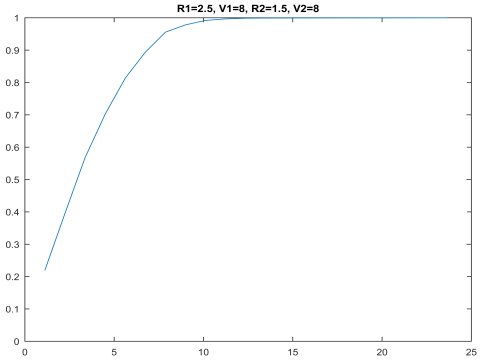

R1=2, V1=8, R2=1.5, V2=8

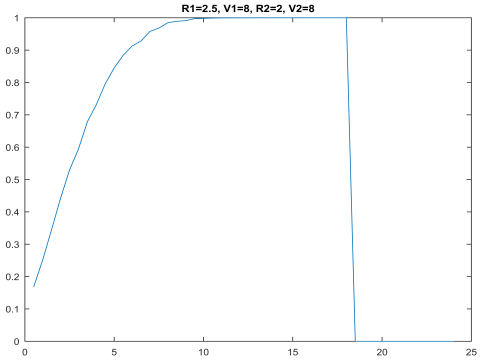

R1=2, V1=8, R2=2, V2=8

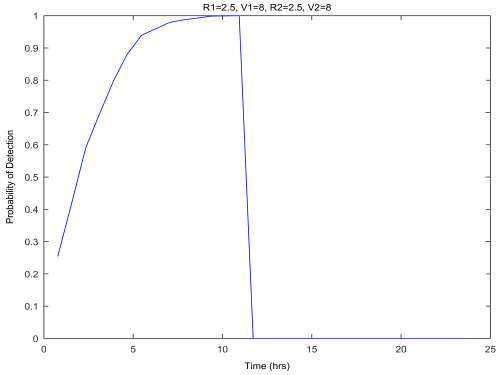

R1=2, V1=8, R2=2.5, V2=8


R1=2.5, V1=8, R2=.5, V2=8

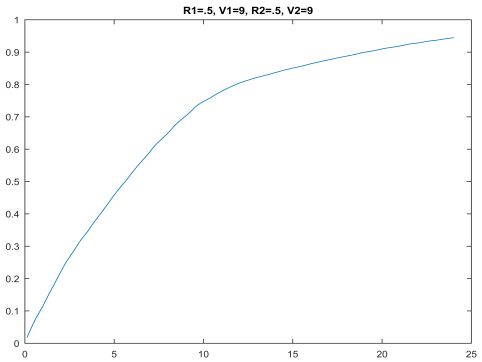

R1=2.5, V1=8, R2=1, V2=8

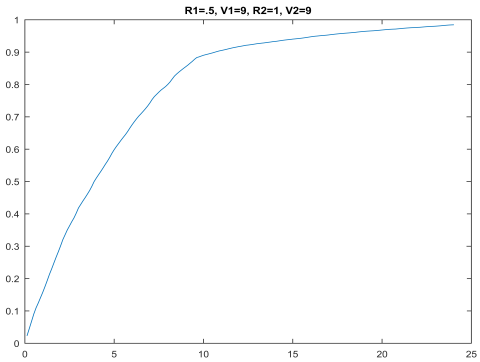

R1=2.5, V1=8, R2=1.5, V2=8

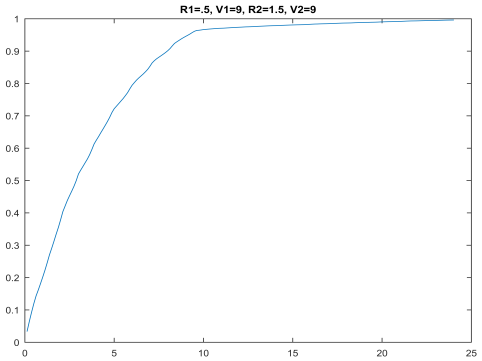

R1=2.5, V1=8, R2=2, V2=8

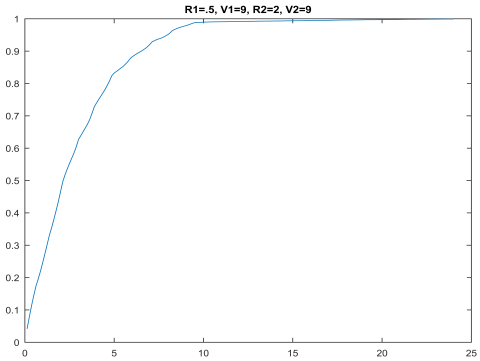

R1=2.5, V1=8, R2=2.5, V2=8
Probability of Detection
Time (hrs)

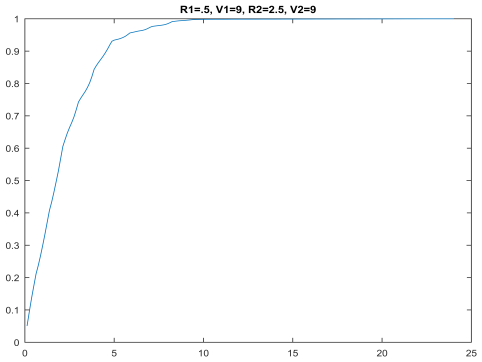
9 kts

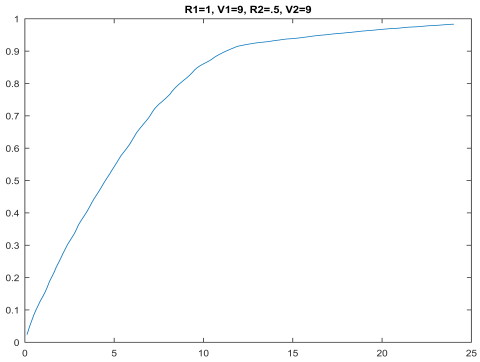

R1=.5, V1=9, R2=.5, V2=9

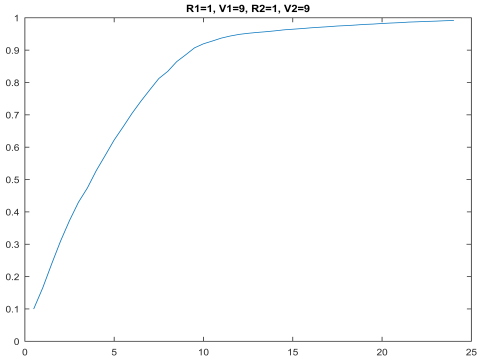

R1=.5, V1=9, R2=1, V2=9

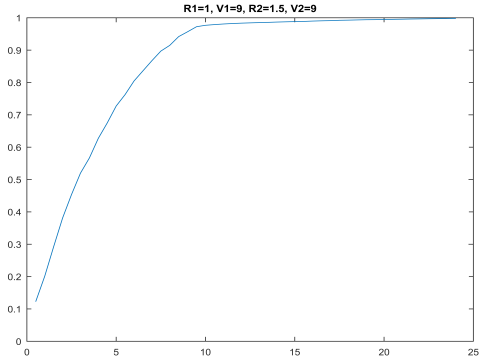

R1=.5, V1=9, R2=1.5, V2=9

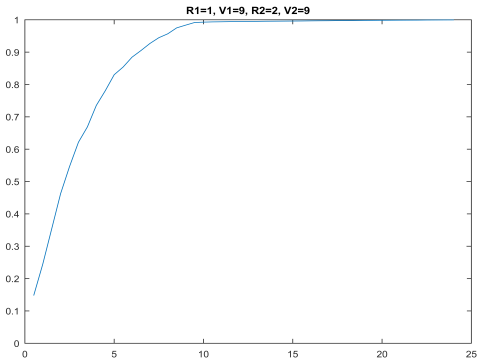

R1=.5, V1=9, R2=2, V2=9

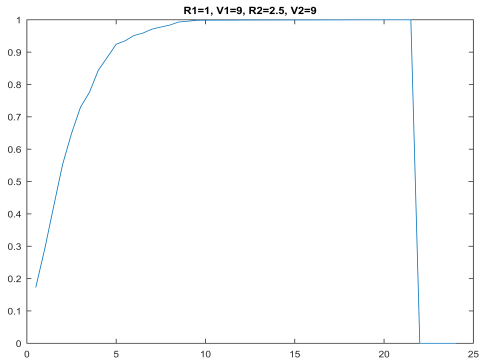

R1=.5, V1=9, R2=2.5, V2=9

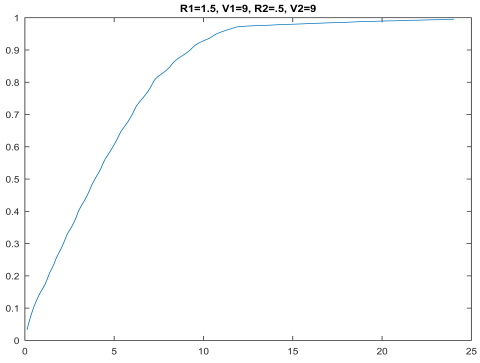

R1=1, V1=9, R2=.5, V2=9

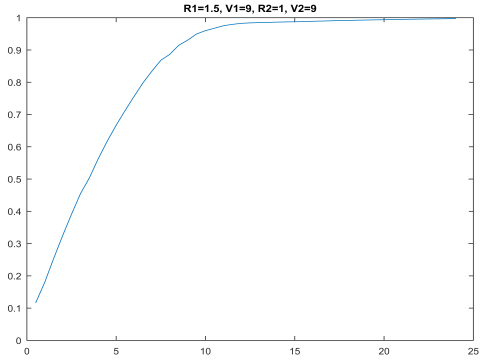

R1=1, V1=9, R2=1, V2=9

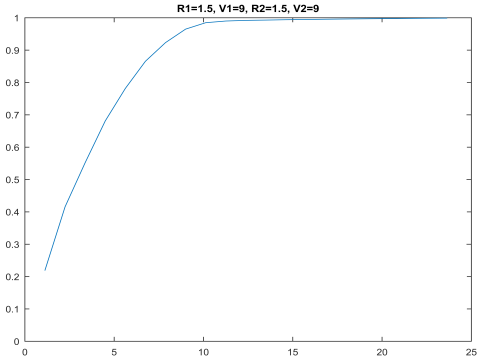

R1=1, V1=9, R2=1.5, V2=9

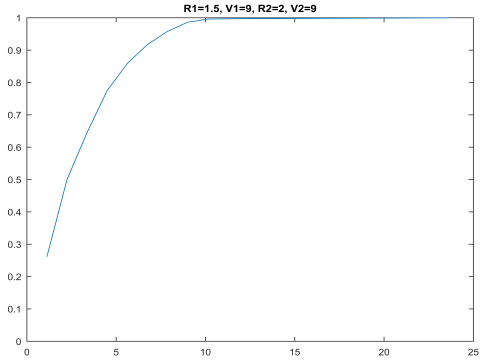

R1=1, V1=9, R2=2, V2=9

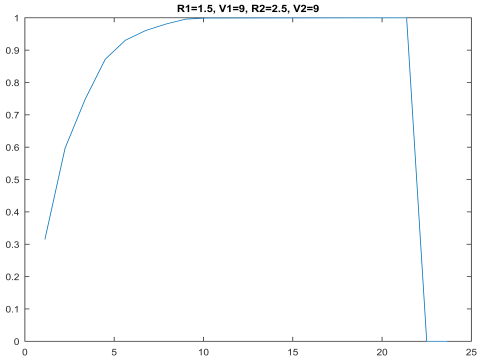

R1=1, V1=9, R2=2.5, V2=9

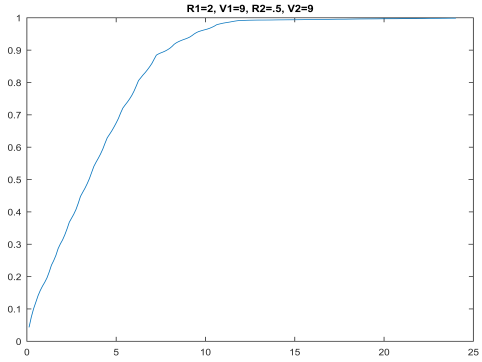

R1=1.5, V1=9, R2=.5, V2=9

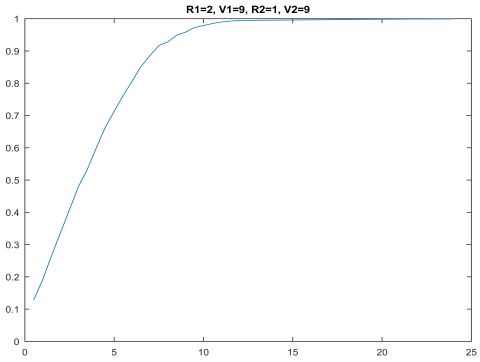

R1=1.5, V1=9, R2=1, V2=9

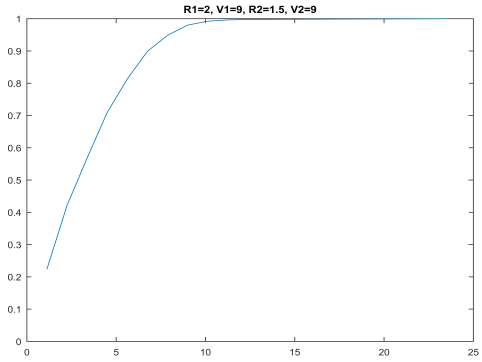

R1=1.5, V1=9, R2=1.5, V2=9

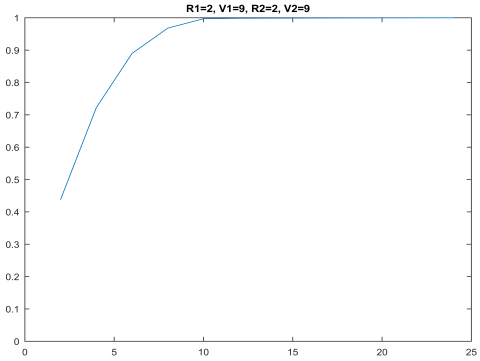

R1=1.5, V1=9, R2=2, V2=9

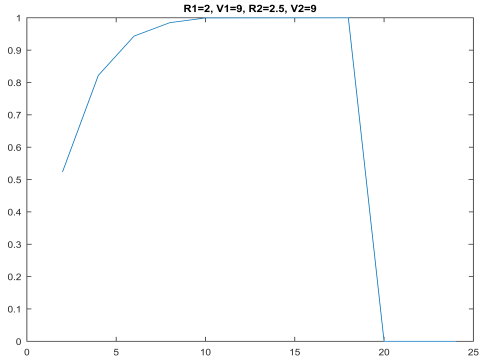

R1=1.5, V1=9, R2=2.5, V2=9

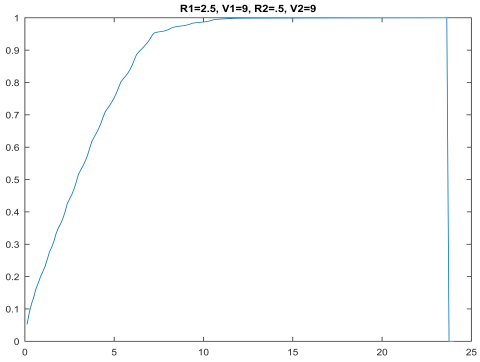

R1=2, V1=9, R2=.5, V2=9

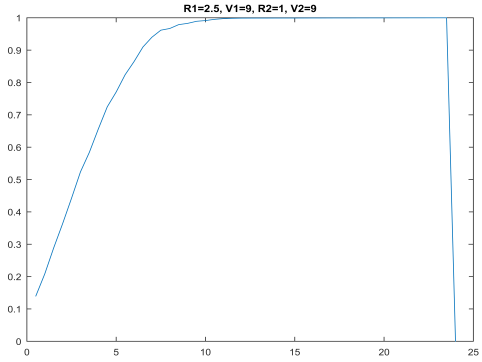

R1=2, V1=9, R2=1, V2=9

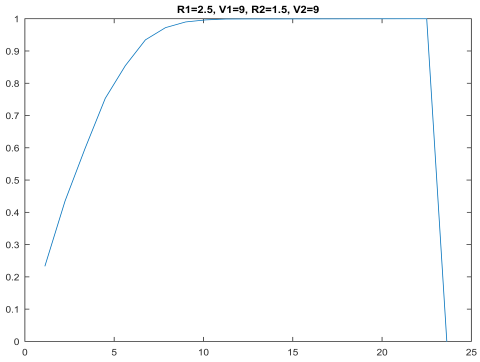

R1=2, V1=9, R2=1.5, V2=9

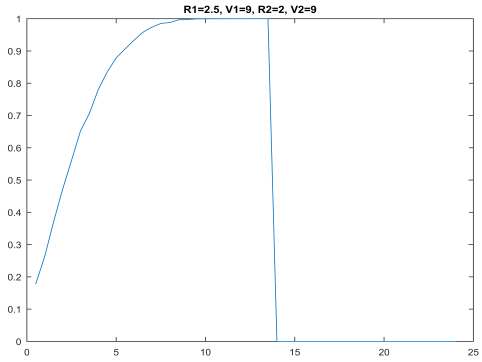

R1=2, V1=9, R2=2, V2=9

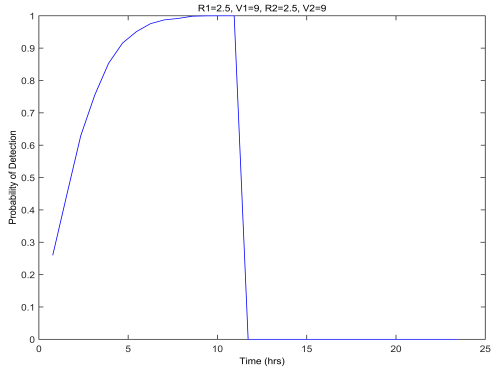

R1=2, V1=9, R2=2.5, V2=9


R1=2.5, V1=9, R2=.5, V2=9

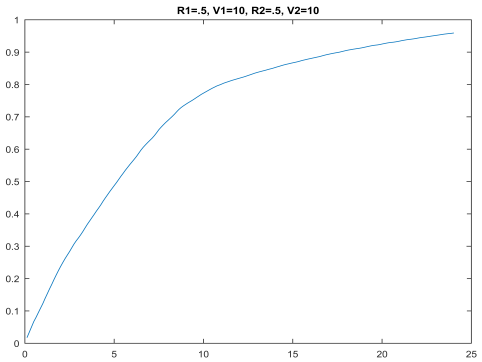

R1=2.5, V1=9, R2=1, V2=9

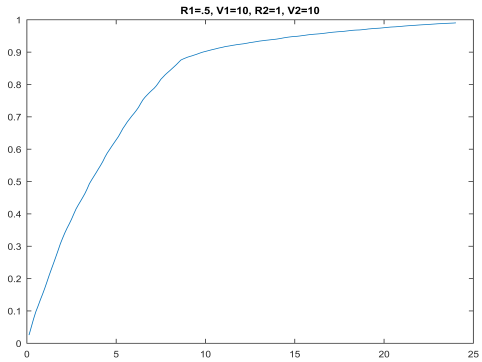

R1=2.5, V1=9, R2=1.5, V2=9

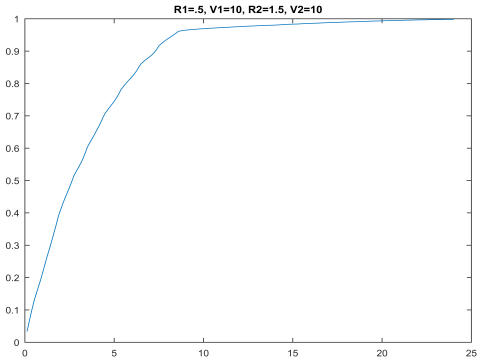

R1=2.5, V1=9, R2=2, V2=9

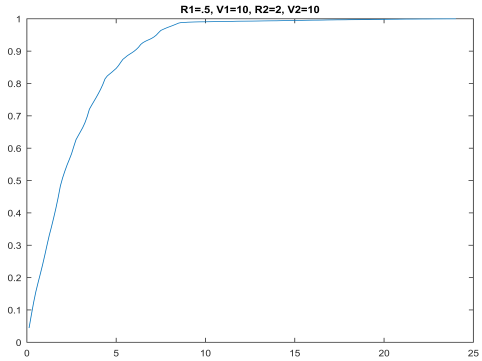

R1=2.5, V1=9, R2=2.5, V2=9
Probability of Detection
Time (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

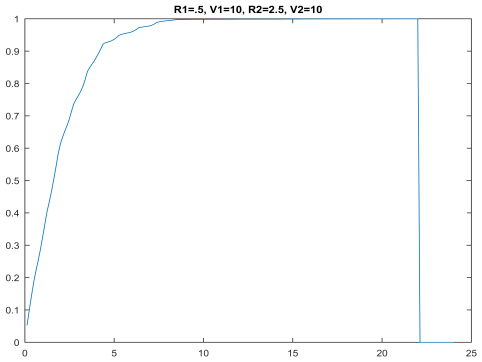
10 kts

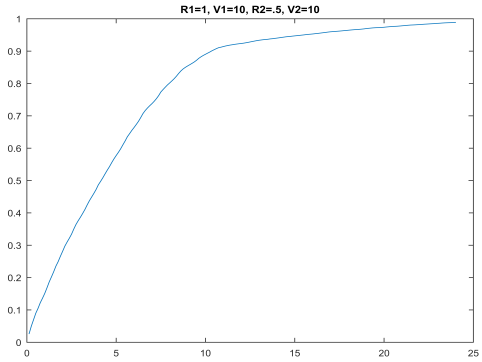

R1=.5, V1=10, R2=.5, V2=10

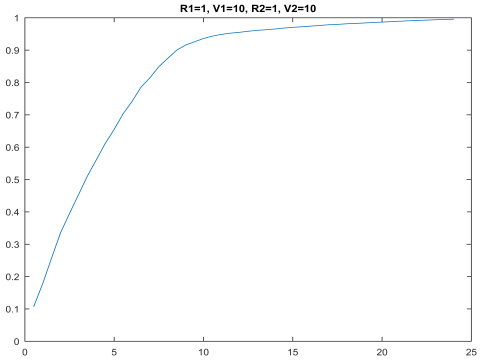

R1=.5, V1=10, R2=1, V2=10

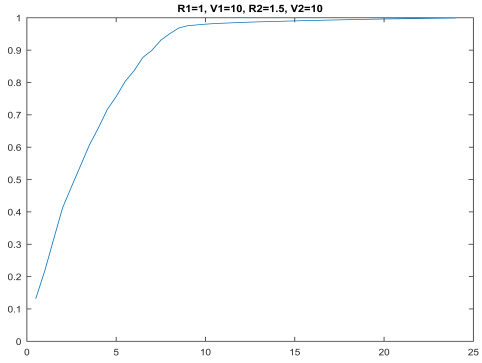

R1=.5, V1=10, R2=1.5, V2=10

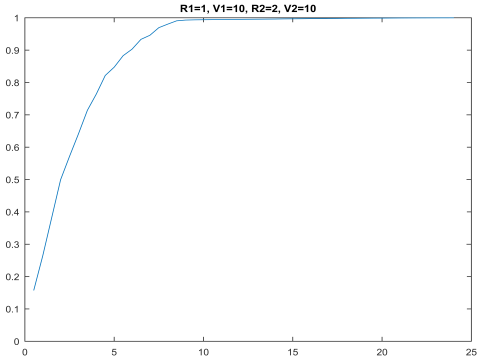

R1=.5, V1=10, R2=2, V2=10

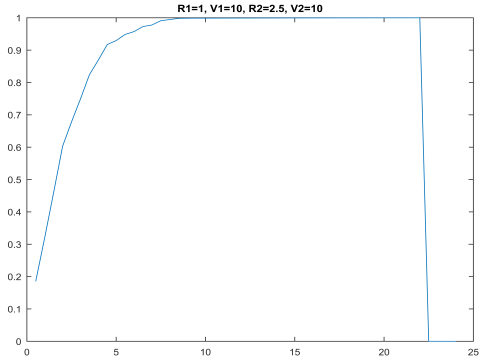

R1=.5, V1=10, R2=2.5, V2=10

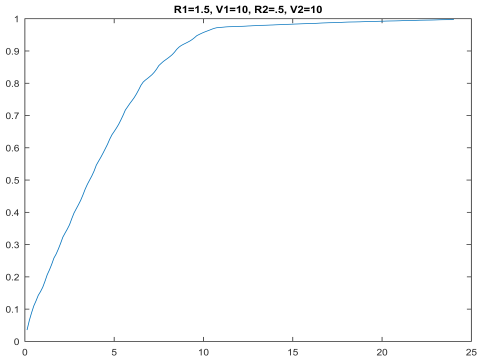

R1=1, V1=10, R2=.5, V2=10

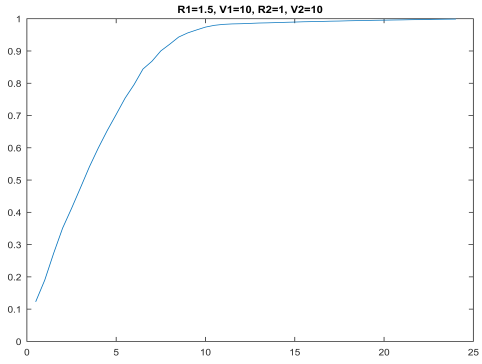

R1=1, V1=10, R2=1, V2=10

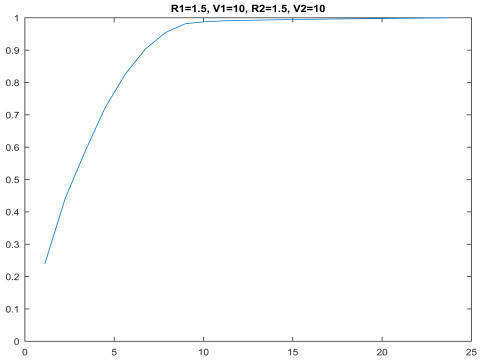

R1=1, V1=10, R2=1.5, V2=10

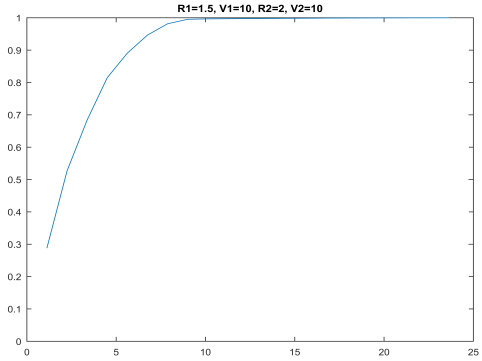

R1=1, V1=10, R2=2, V2=10

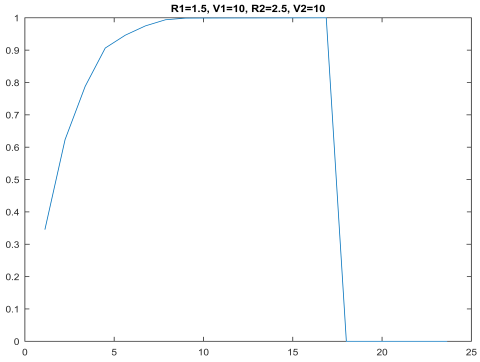

R1=1, V1=10, R2=2.5, V2=10

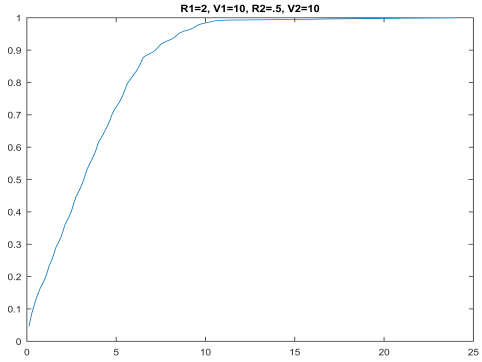

R1=1.5, V1=10, R2=.5, V2=10

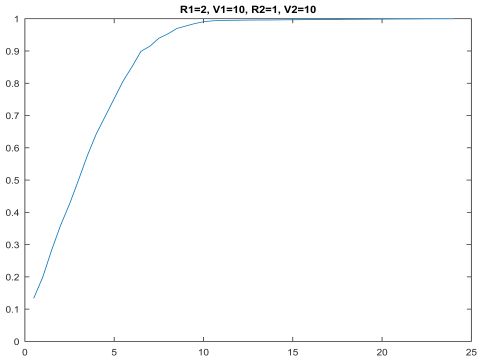

R1=1.5, V1=10, R2=1, V2=10

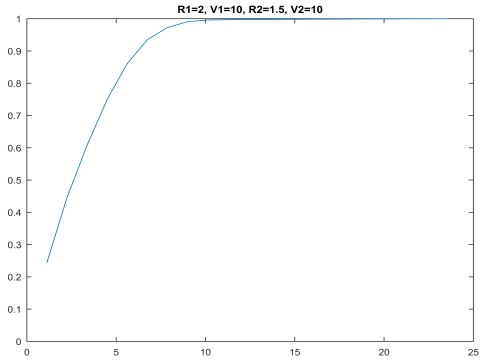

R1=1.5, V1=10, R2=1.5, V2=10

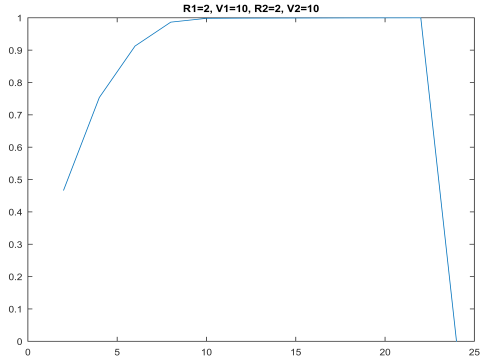

R1=1.5, V1=10, R2=2, V2=10

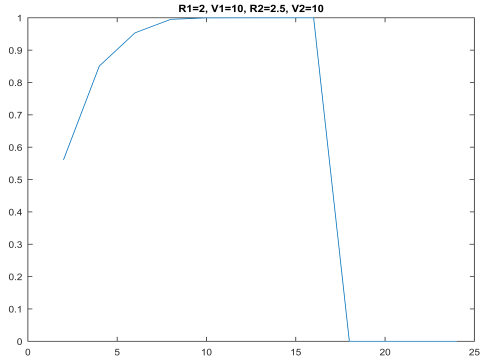

R1=1.5, V1=10, R2=2.5, V2=10

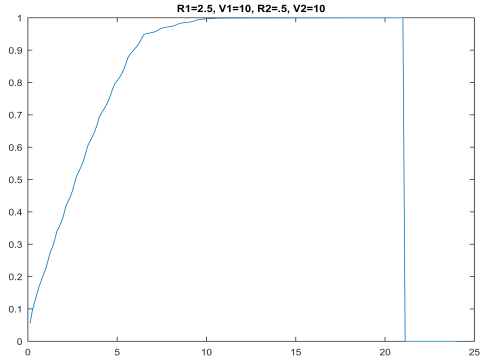

R1=2, V1=10, R2=.5, V2=10

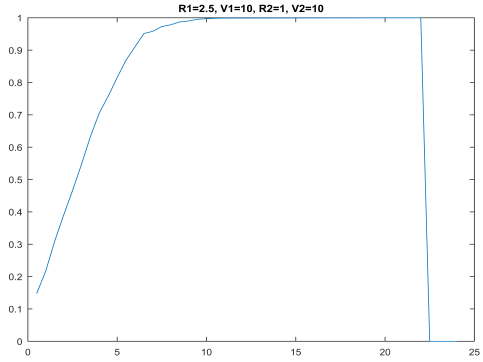

R1=2, V1=10, R2=1, V2=10

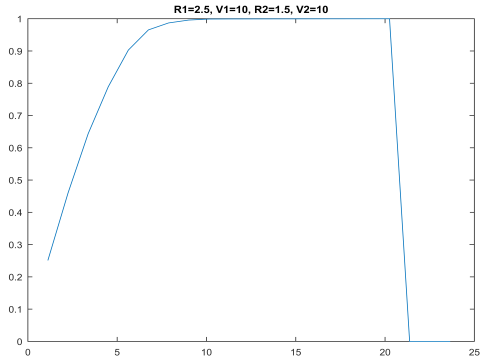

R1=2, V1=10, R2=1.5, V2=10

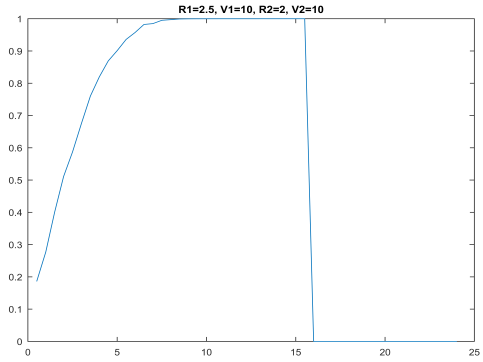

R1=2, V1=10, R2=2, V2=10

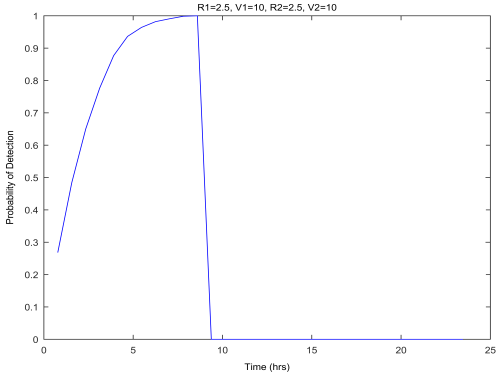

R1=2, V1=10, R2=2.5, V2=10


R1=2.5, V1=10, R2=.5, V2=10


R1=2.5, V1=10, R2=1, V2=10

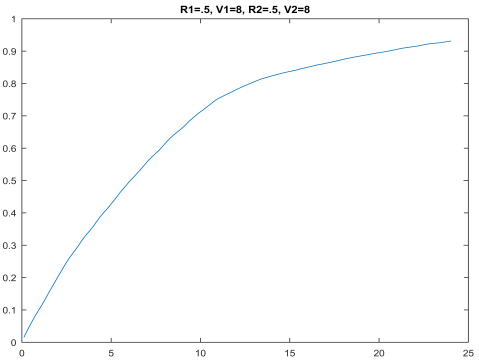

R1=2.5, V1=10, R2=1.5, V2=10

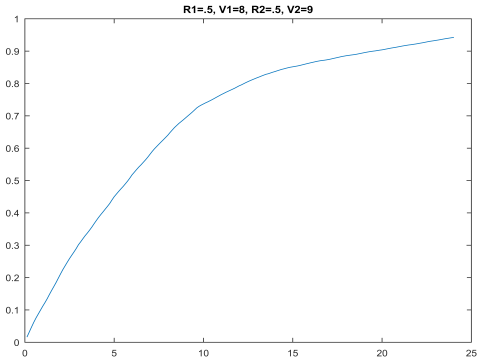

R1=2.5, V1=10, R2=2, V2=10

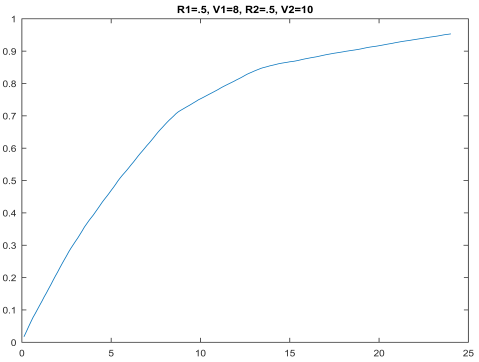

R1=2.5, V1=10, R2=2.5, V2=10
Probability of Detection
Time (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

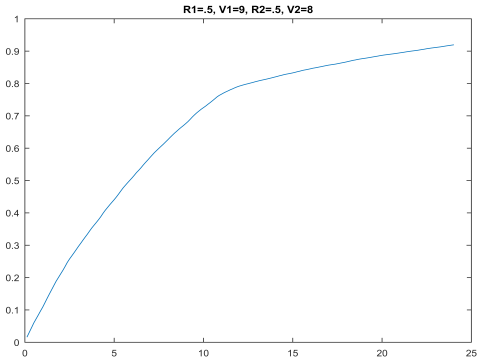
A.3.2 Varying Speed, Constant Search Radii

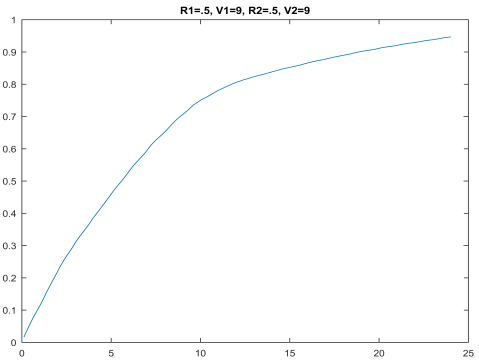
.5nm

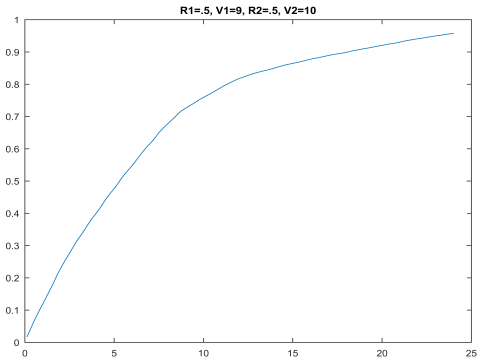


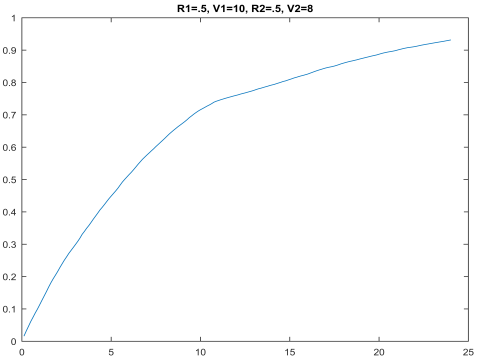


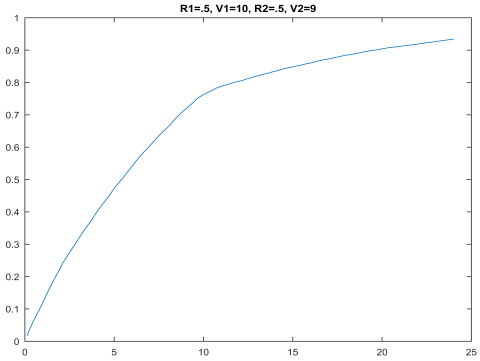


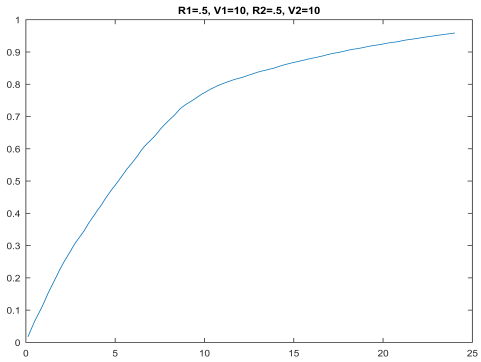




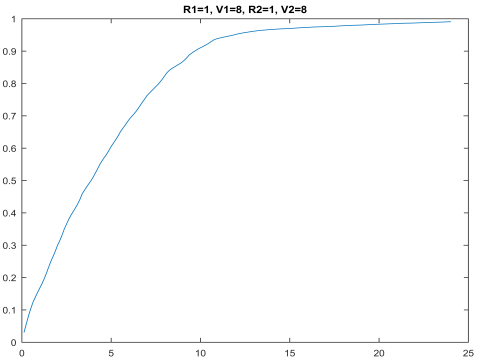


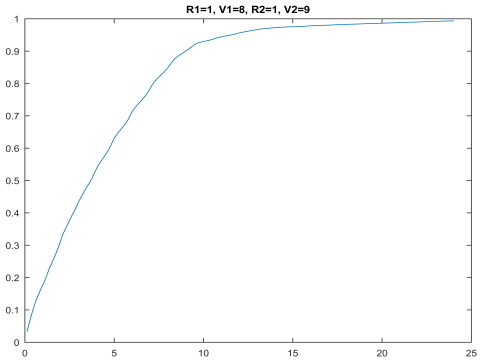

R1=.5, V1=10, R2=.5, V2=8

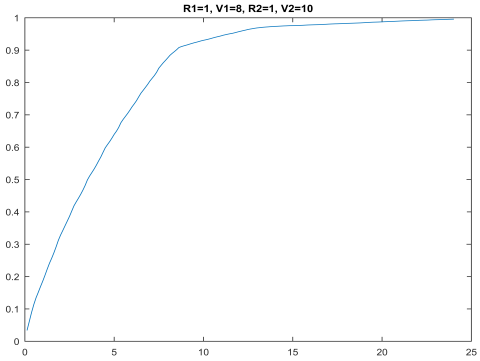

R1=.5, V1=10, R2=.5, V2=9

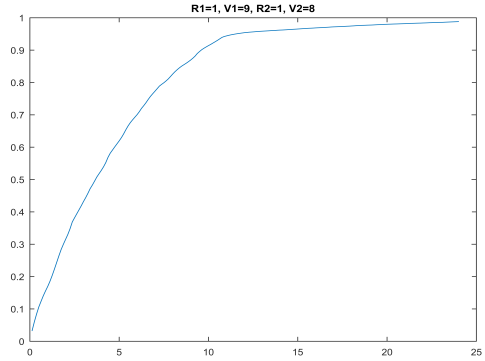

R1=.5, V1=10, R2=.5, V2=10

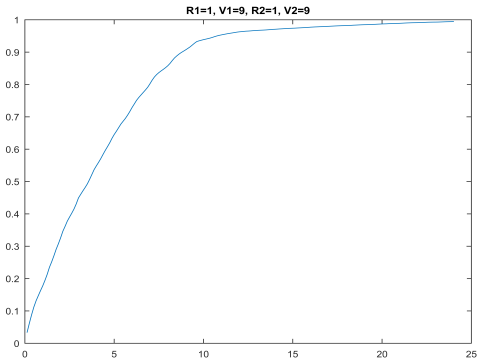
1nm

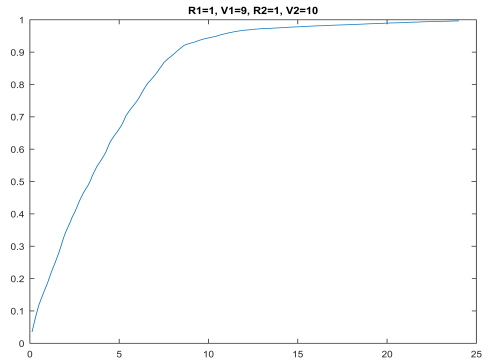

R1=1, V1=8, R2=1, V2=8

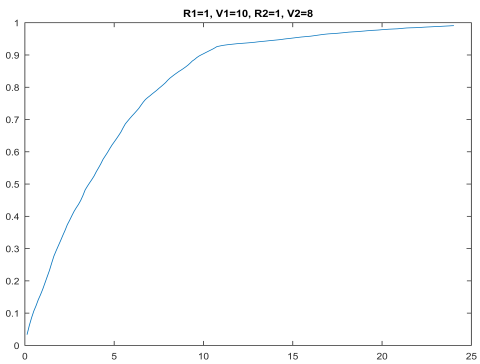

R1=1, V1=8, R2=1, V2=9

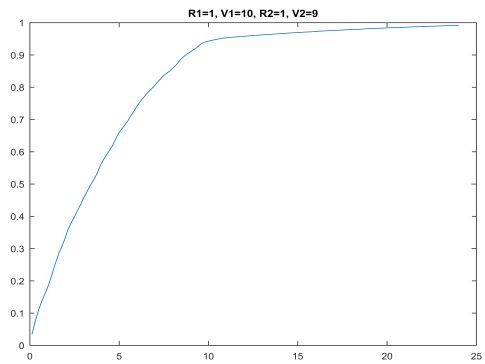

R1=1, V1=8, R2=1, V2=10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

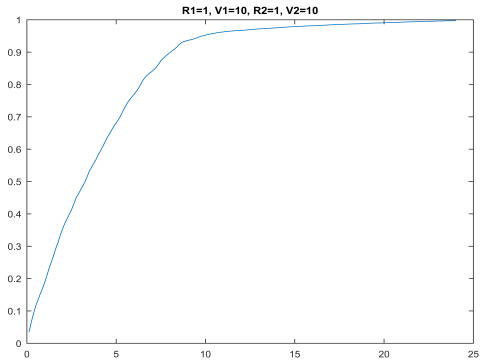

R1=1, V1=9, R2=1, V2=8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25


R1=1, V1=9, R2=1, V2=9
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

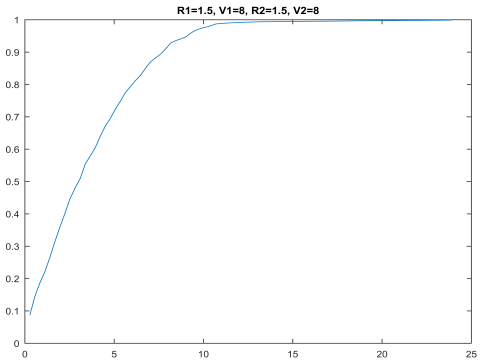

R1=1, V1=9, R2=1, V2=10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

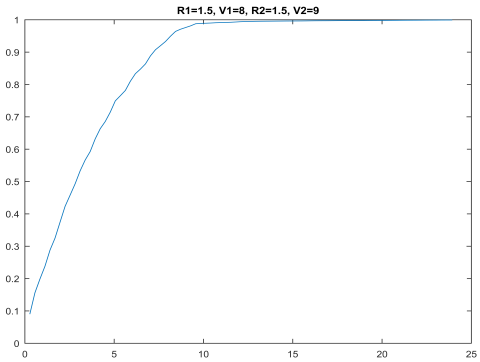

R1=1, V1=10, R2=1, V2=8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

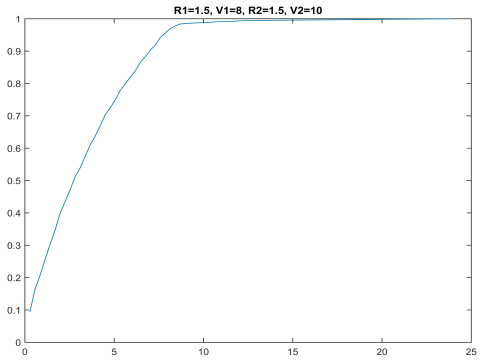

R1=1, V1=10, R2=1, V2=9
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

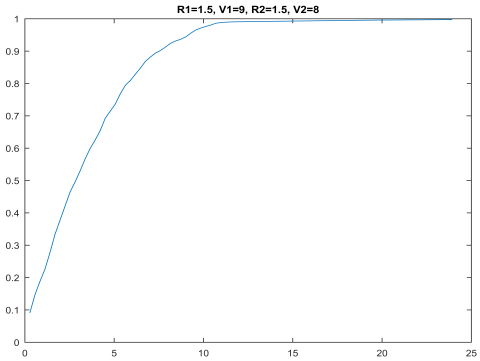

R1=1, V1=10, R2=1, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

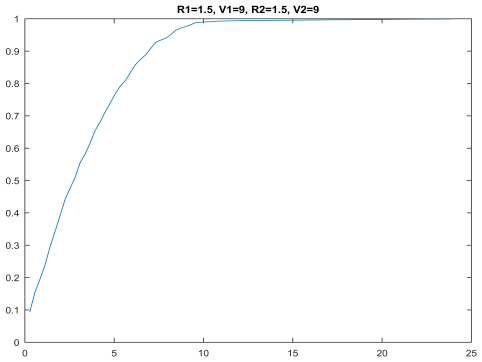
1.5nm

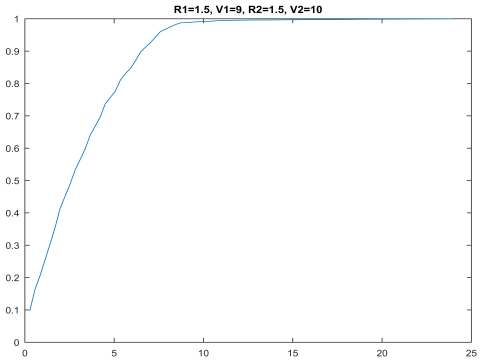

R1=1.5, V1=8, R2=1.5, V2=8

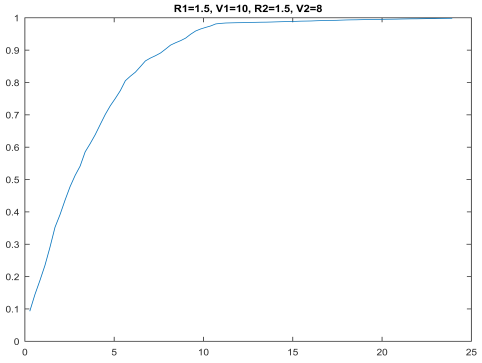

R1=1.5, V1=8, R2=1.5, V2=9

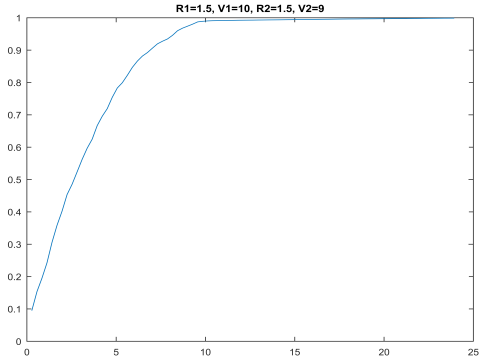

R1=1.5, V1=8, R2=1.5, V2=10

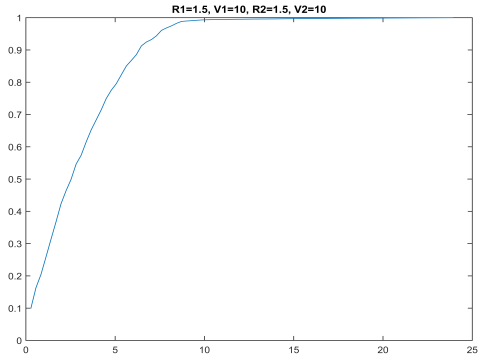

R1=1.5, V1=9, R2=1.5, V2=8


R1=1.5, V1=9, R2=1.5, V2=9

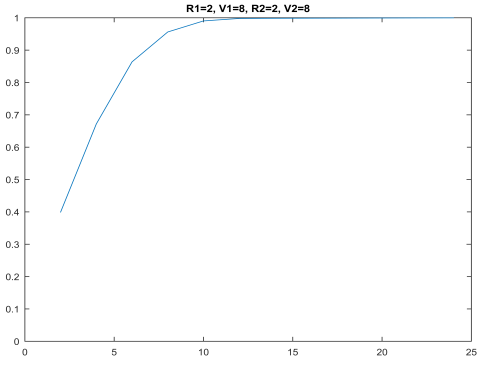

R1=1.5, V1=9, R2=1.5, V2=10

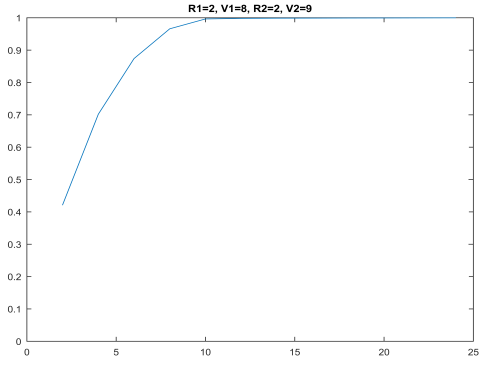

R1=1.5, V1=10, R2=1.5, V2=8

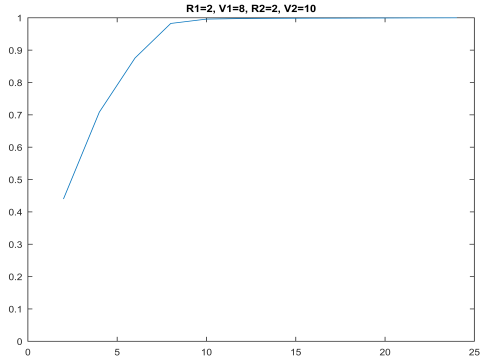

R1=1.5, V1=10, R2=1.5, V2=9

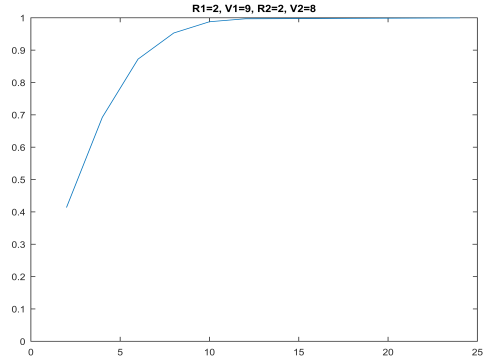

R1=1.5, V1=10, R2=1.5, V2=10

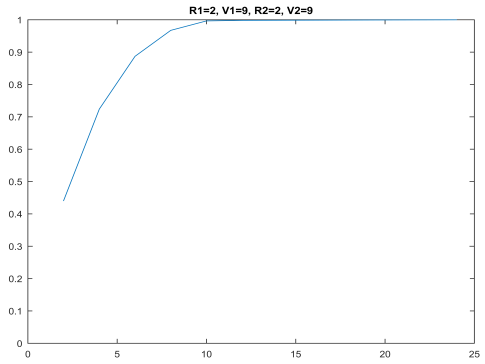
2nm

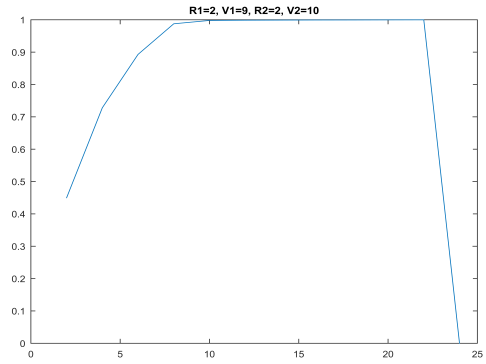

R1=2, V1=8, R2=2, V2=8

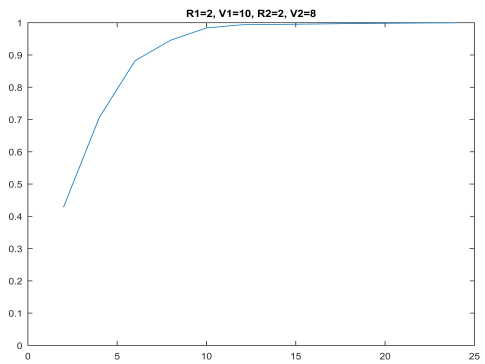

R1=2, V1=8, R2=2, V2=9

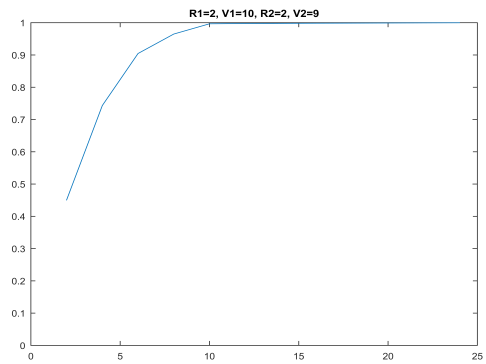

R1=2, V1=8, R2=2, V2=10

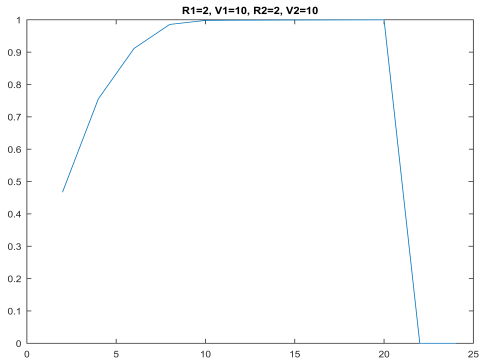

R1=2, V1=9, R2=2, V2=8


R1=2, V1=9, R2=2, V2=9

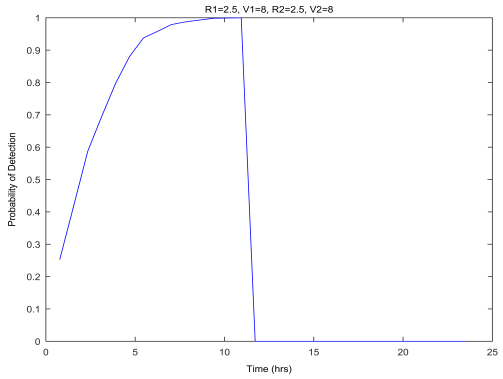

R1=2, V1=9, R2=2, V2=10

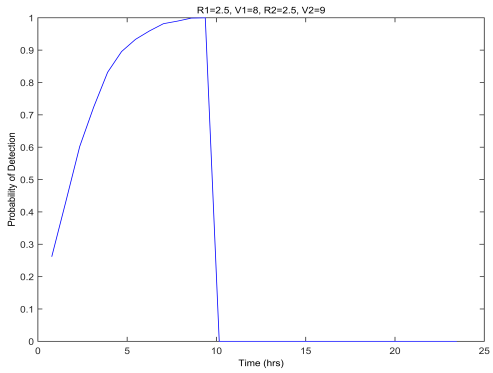

R1=2, V1=10, R2=2, V2=8

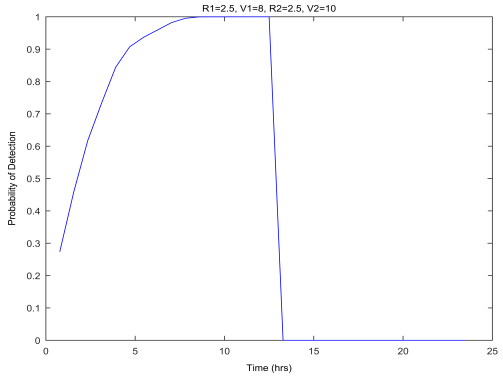

R1=2, V1=10, R2=2, V2=9

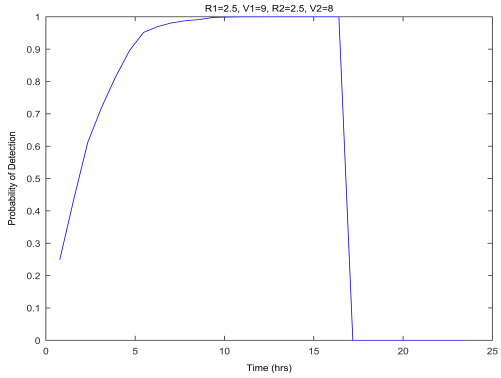

R1=2, V1=10, R2=2, V2=10
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20
25

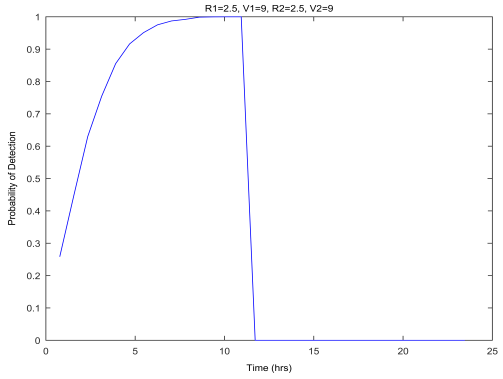
2.5nm

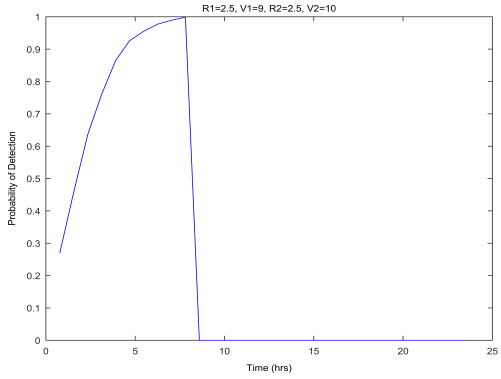

R1=2.5, V1=8, R2=2.5, V2=8
Probability of Detection
Time (hrs)

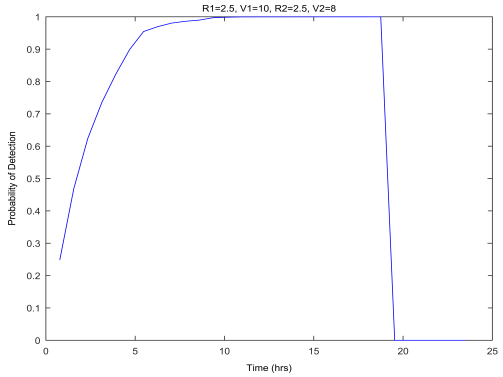

R1=2.5, V1=8, R2=2.5, V2=9
Probability of Detection
Time (hrs)

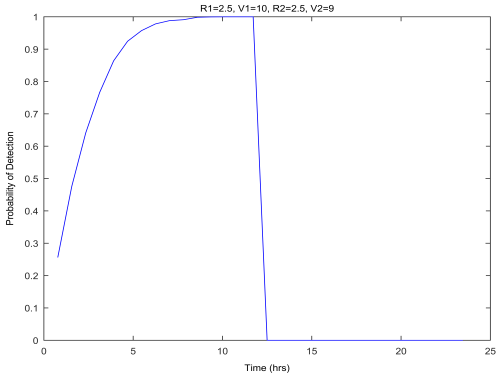

R1=2.5, V1=8, R2=2.5, V2=10
Probability of Detection
Time (hrs)

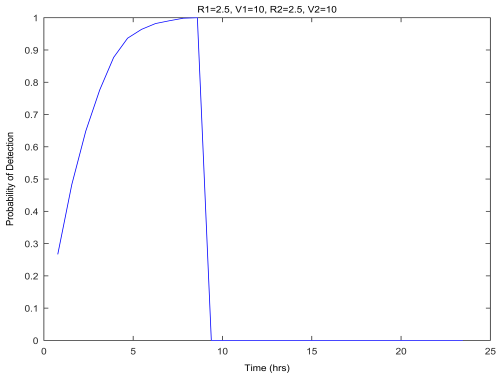

R1=2.5, V1=9, R2=2.5, V2=8
Probability of Detection
Time (hrs)


R1=2.5, V1=9, R2=2.5, V2=9
Probability of Detection
Time (hrs)


R1=2.5, V1=9, R2=2.5, V2=10
Probability of Detection
Time (hrs)


R1=2.5, V1=10, R2=2.5, V2=8
Probability of Detection
Time (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25


R1=2.5, V1=10, R2=2.5, V2=9
Probability of Detection
Time (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25


R1=2.5, V1=10, R2=2.5, V2=10
Probability of Detection
Time (hrs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20
25

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Tables

The following tables represent the amount of time (in hours) each separate scenario takes to reach a 90 percent probability of detection.

B.1 In-Line Ladder Search

B.1.1 Fixed Velocity, Varying Search Radii

Table B.1. Velocity = 8 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=14.84	T=11.94	T=8.93	T=6.09	T=5.03
	1	T=10.78	T=7.13	T=6.12	T=5.50	T=4.63
Ship 1	1.5	T=6.97	T=6.25	T=5.91	T=5.34	T=4.78
	2	T=6.41	T=6.00	T=5.91	T=5.50	T=5.00
	2.5	T=6.16	T=5.88	T=5.63	T=5.50	T=4.69

Table B.2. Velocity = 9kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=13.47	T=10.75	T=8.25	T=5.47	T=4.69
	1	T=9.81	T=6.34	T=5.50	T=4.89	T=4.25
Ship 1	1.5	T=6.25	T=5.63	T=5.34	T=4.78	T=4.22
	2	T=5.72	T=5.38	T=5.06	T=5.00	T=4.50
	2.5	T=5.50	T=5.25	T=5.06	T=5.00	T=4.69

Table B.3. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
Ship 1	.5	T=12.28	T=9.94	T=7.63	T=4.94	T=4.28
	1	T=8.97	T=5.88	T=5.00	T=4.50	T=3.75
	1.5	T=5.66	T=5.00	T=4.78	T=4.50	T=3.66
	2	T=5.16	T=4.88	T=4.78	T=4.50	T=4.00
	2.5	T=4.97	T=4.75	T=4.50	T=4.50	T=3.91

B.1.2 Fixed Search Radius, Varying Velocities

Table B.4. Search Radius = .5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=14.78	T=13.84	T=13.22
	9	T=14.72	T=13.41	T=12.69
	10	T=15.47	T=13.31	T=12.25

Table B.5. Search Radius = 1 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=7.13	T=6.88	T=6.50
	9	T=7.13	T=6.38	T=6.25
	10	T=7.63	T=6.63	T=5.88

Table B.6. Search Radius = 1.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=5.91	T=5.34	T=5.06
	9	T=5.63	T=5.34	T=4.78
	10	T=5.63	T=5.06	T=4.78

Table B.7. Search Radius = 2 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=5.50	T=5.00	T=4.50
	9	T=5.50	T=5.00	T=4.50
	10	T=5.00	T=5.00	T=4.50

Table B.8. Search Radius = 2.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=4.69	T=4.69	T=3.91
	9	T=4.69	T=4.69	T=3.91
	10	T=4.69	T=4.96	T=3.91

B.2 Multi-Path Ladder Search

B.2.1 Fixed Velocity, Varying Search Radii

Table B.9. Velocity = 8 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=15.72	T=10.47	T=8.91	T=8.53	T=8.31
	1	T=9.67	T=8.88	T=8.50	T=8.25	T=8.13
Ship 1	1.5	T=8.53	T=8.25	T=8.16	T=7.88	T=7.88
	2	T=7.38	T=7.13	T=7.03	T=7.00	T=7.00
	2.5	T=6.72	T=6.63	T=6.75	T=6.50	T=7.03

Table B.10. Velocity = 9 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=14.25	T=9.44	T=7.97	T=7.63	T=7.41
	1	T=9.03	T=8.00	T=7.63	T=7.34	T=7.25
Ship 1	1.5	T=7.59	T=7.34	T=7.31	T=7.31	T=7.03
	2	T=6.66	T=6.38	T=6.47	T=6.50	T=6.00
	2.5	T=6.00	T=5.88	T=5.91	T=6.00	T=6.25

Table B.11. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
Ship 1	.5	T=13.00	T=8.75	T=7.12	T=6.88	T=6.69
	1	T=8.38	T=7.25	T=6.88	T=6.63	T=6.63
	1.5	T=6.88	T=6.63	T=6.75	T=6.47	T=6.47
	2	T=6.06	T=5.75	T=5.63	T=6.00	T=5.50
	2.5	T=5.44	T=5.34	T=5.34	T=5.50	T=5.47

B.2.2 Fixed Search Radius, Varying Velocities

Table B.12. Search Radius = .5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=15.72	T=16.16	T=15.84
	9	T=14.53	T=14.25	T=14.38
	10	T=13.56	T=13.25	T=13.00

Table B.13. Search Radius = 1 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=8.88	T=8.88	T=9.38
	9	T=8.25	T=8.00	T=8.00
	10	T=7.75	T=7.34	T=7.25

Table B.14. Search Radius = 1.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=8.16	T=7.88	T=8.16
	9	T=7.59	T=7.31	T=7.03
	10	T=6.75	T=6.75	T=6.75

Table B.15. Search Radius = 2 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=7.00	T=7.00	T=7.00
	9	T=6.50	T=6.50	T=6.50
	10	T=6.00	T=6.00	T=6.00

Table B.16. Search Radius = 2.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=7.03	T=7.03	T=7.03
	9	T=6.25	T=6.25	T=6.25
	10	T=5.47	T=5.47	T=5.47

B.3 In-Line Spiral Search

B.3.1 Fixed Velocity, Varying Search Radii

Table B.17. Velocity = 8 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=20.63	T=11.63	T=8.88	T=7.13	T=5.13
	1	T=15.50	T=10.50	T=8.50	T=7.00	T=5.50
Ship 1	1.5	T=10.25	T=9.50	T=9.00	T=7.88	T=5.63
	2	T=8.25	T=8.00	T=7.88	T=8.00	T=6.00
	2.5	T=7.13	T=7.00	T=7.88	T=6.00	T=5.47

Table B.18. Velocity = 9 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=18.88	T=10.63	T=8.13	T=6.63	T=4.63
	1	T=11.34	T=9.50	T=8.00	T=6.50	T=5.00
Ship 1	1.5	T=9.34	T=8.50	T=7.88	T=6.75	T=5.63
	2	T=7.88	T=7.50	T=7.88	T=8.00	T=6.00
	2.5	T=6.50	T=6.50	T=6.75	T=5.50	T=4.69

Table B.19. Velocity = 10 kts

		Ship 2				
	Search Radius (nm)	.5	1	1.5	2	2.5
	.5	T=17.75	T=9.88	T=7.38	T=6.13	T=4.25
	1	T=10.34	T=9.00	T=7.50	T=6.00	T=4.50
Ship 1	1.5	T=8.50	T=7.50	T=6.75	T=6.75	T=4.50
	2	T=7.25	T=7.00	T=6.75	T=6.00	T=6.00
	2.5	T=6.00	T=6.00	T=5.63	T=5.00	T=4.69

B.3.2 Fixed Search Radius, Varying Velocities

Table B.20. Search Radius = .5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=20.62	T=19.62	T=18.25
	9	T=21.63	T=18.88	T=18.13
	10	T=21.13	T=19.63	T=17.63

Table B.21. Search Radius = 1 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=9.75	T=9.13	T=8.50
	9	T=9.63	T=8.88	T=8.25
	10	T=9.88	T=8.75	T=8.13

Table B.22. Search Radius = 1.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=7.88	T=7.31	T=7.03
	9	T=7.59	T=7.03	T=6.75
	10	T=7.88	T=7.03	T=6.47

Table B.23. Search Radius = 2 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=6.50	T=6.50	T=6.50
	9	T=6.00	T=6.00	T=6.00
	10	T=6.00	T=6.00	T=5.50

Table B.24. Search Radius = 2.5 nm

		Ship 2		
	Velocity (kts)	8	9	10
Ship 1	8	T=5.47	T=5.47	T=4.69
	9	T=5.47	T=4.69	T=4.69
	10	T=5.47	T=4.69	T=4.69

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:
Code

The following sections provide the code utilized to achieve the previous results.

C.1 Spiral Code

```
%
% Need to input ns !!!
ns=5;
s0=3*7; % two searchers are separated by s0 on the path (3*Lx)
%
tic
%
% Calculate sv xv
[sv, xv]=sspiral_calc_sx(ns);
%
Dt=1; % diffusion coefficient of targets
%
N=100000;    % number of repeats
Lx=7;  % half width
Ly=7;  % half height
R1=2.5;   % radius of searcher 1
R2=2.5;   % radius of searcher 2
R=min(R1,R2);
T=24;  % final time in computation
%
Lxs=Lx;   % reflecting boundaries for searcher
Lys=Ly;   % reflecting boundaries for searcher
dt=(R/4)^2/(2*Dt);
dt=dt/4;
kt=round(T/dt);
cft=sqrt(2*Dt*dt);  % numerical coefficient of targets

% searcher one
x_vec=sspiral_xy(0, ns, sv, xv);
```

```
xs0=x_vec(1);   % initial position of searcher one
ys0=x_vec(2);
v1=10; %velocity of searcher one

% searcher two
x_vec=sspiral_xy(s0, ns, sv, xv);
xsb0=x_vec(1);  % initial position of searcher
ysb0=x_vec(2);
v2=10; %velocity of searcher two

% distribute N targets uniformly outside the searcher
N2=round(1.5*N);
xt=2*Lx*(rand(N2,1)-0.5);
yt=2*Ly*(rand(N2,1)-0.5);
ind=find( ((xt-xs0).^2+(yt-ys0).^2 > R1^2) & ...
 ((xt-xsb0).^2+(yt-ysb0).^2 > R2^2)   );
xt=xt(ind(1:N));
yt=yt(ind(1:N));
xs=xs0*ones(N,1);
ys=ys0*ones(N,1);
xsb=xsb0*ones(N,1);
ysb=ysb0*ones(N,1);
%
n=N;  % current number of remaining systems
pa=zeros(kt,1); % probability of escaping
for k=1:kt,
  x_vec=sspiral_xy(k*dt*v1, ns, sv, xv); %searcher one
  xs=0*xs+x_vec(1);
  ys=0*ys+x_vec(2);
  x_vec=sspiral_xy(k*dt*v2+s0, ns, sv, xv); %searcher two
  xsb=0*xsb+x_vec(1);
  ysb=0*ysb+x_vec(2);
  %
  xt=xt+cft*randn(n,1);
  xt=abs(xt+Lx)-Lx;
  xt=Lx-abs(Lx-xt);
  %
  yt=yt+cft*randn(n,1);
  yt=abs(yt+Ly)-Ly;
  yt=Ly-abs(Ly-yt);
```

```
  %
  ind=find( ((xt-xs).^2+(yt-ys).^2 > R1^2) &...
      ((xt-xsb).^2+(yt-ysb).^2 > R2^2) );
  n=size(ind,1);
  pa(k)=1-n/N;
  if n<10,
    break
  end
  xt=xt(ind);
  yt=yt(ind);
  xs=xs(ind);
  ys=ys(ind);
  xsb=xsb(ind);
  ysb=ysb(ind);
end
t2=toc;
dn=16;
ta=[dn:dn:kt]*dt;
pa=pa(dn:dn:kt);
%
eval(['save pa_ns',num2str(ns)])
%
figure
plot(ta,pa)
ylim([0 1])
xlabel('Time (hrs)')
ylabel('Probability of Detection')
title('R1=2.5, V1=10, R2=2.5, V2=10')
%

a = find(pa>.9,1)
ta(a)
```

C.1.1 Spiral Calculation Fuction

```
function [sv, xv]=sspiral_calc_sx(ns)
%
n=ns;
```

```
dx=zeros(4*n,2);
ds=zeros(4*n,1);
%
for k=1:n
  dx(4*k-3,:)=[2*k-1, 0];
  dx(4*k-2,:)=[0, 2*k-1];
  dx(4*k-1,:)=[-2*k, 0];
  dx(4*k,:)=[0, -2*k];
  ds(4*k-3)=2*k-1;
  ds(4*k-2)=2*k-1;
  ds(4*k-1)=2*k;
  ds(4*k)=2*k;
end
ds2=ds(4*n:-1:1);
ds2=[ds2(1)-0.5; ds2(2)-0.5; ds2(3:4*n); 0.5; 0.5];
%
dx2=-[dx(4*n:-1:1,2), dx(4*n:-1:1,1)];
dx2=[dx2(1,:)-[0.5,0]; dx2(2,:)-[0,0.5]; dx2(3:4*n,:); [0.5,0]; [0, 0.5]];
%
sv=[0; cumsum([ds; ds2])];
xv=[0, 0; cumsum([dx; dx2])];
%
%
```

C.1.2 Spiral Movement Fuction

```
function [x]=sspiral_xy(sd, ns, sv, xv)
%
% sd: arclength before normalization by d
%
d=7/(ns+0.25);
s0=4*ns^2+2*ns;
%
ir=floor((sd/d)/(2*s0));
y=(sd/d)-ir*(2*s0);
%
i=find(sv(1:end-1)<=y & sv(2:end)>y);
cf=(sv(i+1)-y)/(sv(i+1)-sv(i));
```

```
xc=cf*xv(i,:)+(1-cf)*xv(i+1,:);
%
ang=ir*pi/2;
xc=xc*[cos(ang), sin(ang); -sin(ang), cos(ang)];
x=d*xc;
%
%
```

C.2 Ladder Code

```
%
global Lx;
%
% Need to input ns !!!
%
ns=5;  % try ns=11
R1=2; % radius of searcher one
R2=2; % radius of searcher two
v1=10;  % velocity of the searcher one
v2=10; % velocity of the searcher two
%
tic
rng('shuffle')
%
% Calculate sv xv
[sv, xv]=mowing_calc_sx(ns);
%
Dt=1; % diffusion coefficient of targets
%
N=100000;    % number of repeats
Lx=7;  % half width
Ly=7;  % half height
T=24;  % final time in computation
s0=40*Lx; % two searchers are separated by s0 on the path
%
Lxs=Lx;   % reflecting boundaries for searcher
Lys=Ly;   % reflecting boundaries for searcher
R=min(R1,R2);
```

```
dt=(R/8)^2/(2*Dt);
%dt=dt/4;
kt=round(T/dt);
cft=sqrt(2*Dt*dt);  % numerical coefficient of targets
%
% initial position of searcher 1
x_vec=mowing_xy(0,ns,sv,xv);
xs0=x_vec(1);
ys0=x_vec(2);
% initial position of searcher 2
x_vec=mowing_xy(s0,ns,sv,xv);
xsb0=x_vec(1);
ysb0=x_vec(2);
% distribute N targets uniformly outside the searcher
N2=round(1.2*N);
xt=2*Lx*(rand(N2,1)-0.5);
yt=2*Ly*(rand(N2,1)-0.5);
ind=find( (xt-xs0).^2+(yt-ys0).^2 > R1^2 &...
   (xt-xsb0).^2+(yt-ysb0).^2 > R2^2 );
xt=xt(ind(1:N));
yt=yt(ind(1:N));
xs=xs0*ones(N,1);
ys=ys0*ones(N,1);
xsb=xsb0*ones(N,1);
ysb=ysb0*ones(N,1);
%
n=N;  % current number of remaining systems
pa=zeros(kt,1); % probability of detection
for k=1:kt,
  % searcher one moves along the trajectory
  x_vec=mowing_xy(k*dt*v1, ns, sv, xv);
  xs=0*xs+x_vec(1);
  xs=abs(xs+Lxs)-Lxs;
  xs=Lxs-abs(Lxs-xs);
  %
  ys=0*ys+x_vec(2);
  ys=abs(ys+Lys)-Lys;
  ys=Lys-abs(Lys-ys);
  %
  % searcher two moves along the trajectory
```

```
  x_vec=mowing_xy(s0+k*dt*v2, ns, sv, xv);
  xsb=0*xsb+x_vec(1);
  xsb=abs(xsb+Lxs)-Lxs;
  xsb=Lxs-abs(Lxs-xsb);
  %
  ysb=0*ysb+x_vec(2);
  ysb=abs(ysb+Lys)-Lys;
  ysb=Lys-abs(Lys-ysb);
  %
  xt=xt+cft*randn(n,1);
  xt=abs(xt+Lx)-Lx;
  xt=Lx-abs(Lx-xt);
  %
  yt=yt+cft*randn(n,1);
  yt=abs(yt+Ly)-Ly;
  yt=Ly-abs(Ly-yt);
  %
  ind=find( (xt-xs).^2+(yt-ys).^2 > R1^2 &...
     (xt-xsb).^2+(yt-ysb).^2 > R2^2 );
  n=size(ind,1);
  pa(k)=1-n/N;
  if n<10,
    break
  end
  xt=xt(ind);
  yt=yt(ind);
  xs=xs(ind);
  ys=ys(ind);
  xsb=xsb(ind);
  ysb=ysb(ind);
end
t2=toc;
dn=16;
ta=[dn:dn:kt]*dt;
pa=pa(dn:dn:kt);
%
eval(['save pa_1',num2str(ns)])
%
figure
plot(ta,pa,'k-')
```

```
ylim([0 1])
xlabel('Time (hrs)')
ylabel('Probability of Detection')
title('R1=2, V1=10, R2=2, V2=10')
%
a = find(pa>.9,1)
ta(a)
```

C.2.1 Ladder Calculation Function

```
function [sv, xv]=mowing_calc_sx(ns)
%
n=ns;   % number of scans
d=1;
L=(n-1);
%
dx=zeros(4*n-1,2);
ds=zeros(4*n-1,1);
%
for k=1:n
  dx(2*k-1,:)=L*[-(-1)^k, 0];
  dx(2*k,:)=d*[0,1];
  ds(2*k-1)=L;
  ds(2*k)=d;
end
dx(2*n,:)=d*[0, -1/2];
ds(2*n)=d/2;
for k=n+1:2*n-1
  dx(2*k-1,:)=L*[-(-1)^k, 0];
  dx(2*k,:)=d*[0,-1];
  ds(2*k-1)=L;
  ds(2*k)=d;
end
dx(2*(2*n-1),:)=d*[0, -1/2];
ds(2*(2*n-1))=d/2;
dx(2*(2*n)-1,:)=L*[-1, 0];
ds(2*(2*n)-1)=L;
%
```

```
sv=[0; cumsum(ds)];
xv=[0, 0; cumsum(dx)];
%
%
```

C.2.2 Ladder Movement Function

```
function [x]=mowing_xy(sd, ns, sv, xv)
%
% sd: arclength before normalization by d
global Lx;
%
cs=2*Lx/(ns-0.5);
s0=sv(end);
%
ir=floor((sd/cs)/s0);
y=(sd/cs)-ir*s0;
%
i=find(sv(1:end-1)<=y & sv(2:end)>y);
cf=(sv(i+1)-y)/(sv(i+1)-sv(i));
xc=cf*xv(i,:)+(1-cf)*xv(i+1,:);
%
x=cs*(xc+[-1,-1]*(ns-1)/2);
if mod(ir,2)==1,
  x=[x(2), x(1)];
end
%
%
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] M. Atkinson, "Oa3602 search theory and detection," class notes for OA3602, Dept. of Operations Reasearch, Naval Postgraduate School, Monterey, CA, spring 2015.

[2] B. Bagnarelli. (2016, Jul. 12). Testing the Rule of Law in the South China Sea. *The New York Times*. [Online]. Available: http://www.nytimes.com/2016/07/13/opinion/testing-the-rule-of-law-in-the-south-china-sea.html?_r=0

[3] R. Beckhusen. (2013, Jul. 5). Search theory and big data: Applying the math that sank U-boats to today's intel problems. *Defense News*. [Online]. Available: http://www.defensenews.com/article/20130705/C4ISR02/307050013/

[4] B. O. Koopman, "A theoretical basis for method of search and screening," DTIC Document, Tech. Rep., 1946.

[5] V. Sozen, "Optimal deployment of unmanned aerial vehicles for border surveillance," M.S. thesis, Dept. Ops. Research, NPS, 1 University Cir, Monerey, CA, 2014.

[6] M. Atkinson, M. Kress, and R.-J. Lange, "When is information sufficient for action? search with unreliable yet informative intelligence," *Operations Research*, vol. 64, no. 2, pp. 315–328, 2016.

[7] H. Wang and H. Zhou, "Computational studies on detecting a diffusing target in a square region by a stationary or moving searcher," *American Journal of Operations Research*, vol. 5, no. 02, p. 47, 2015.

[8] A. R. Washburn, *Search and detection*. Institute for Operations Research and the Management Sciences, 2002.

[9] Forecast International, Newtown, CT. (2002). *SQR-19 TACTAS - Archived 11/2003*. [Online]. Available: https://www.forecastinternational.com/archive/disp_old_pdf.cfm?ARC_ID=885. Accessed Aug. 21, 2016.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California